UNIVERSIDADE DE LISBOA
FACULDADE DE CIÊNCIAS
DEPARTAMENTO DE BIOLOGIA ANIMAL

African Marine Invertebrate Data Analysis and Imaging: a Dataset and Digital Platform for Research, Education and Outreach

Henrique Martins Espada Niza

Mestrado em Bioinformática e Biologia Computacional

Dissertação orientada por:
Alexandra Marçal Correia e Luís Filipe Lopes

2019

Acknowledgments and Dedications

It is with great joy and pleasure that I express my gratitude for the unconditional support and dedication of those involved in the realization of this dissertation, without whom none of this would be possible.

To my two dissertation advisors, Alexandra Marçal Correia and Luís Filipe Lopez, I wish to thank you for being such great supporters throughout this entire project. They've made a point of helping in every way possible, and not once were they too busy to assist, no matter the subject.

To my colleague and working partner Marta Bento for being a lovely person, always willing to help, and a pleasure to work with.

To Alexandra Cartaxana for the sympathy and support given with her knowledge of the biological aspect of this project.

To Leonor Brites Soares and David Harper for providing essential assistance throughout the entire geocoding process.

To Luiz Coutinho for providing much-needed support in the field of GIS and web mapping.

To all the Master course professors and coordinators Octávio Paulo, Francisco Couto and Francisco Dionísio, through their lectures, direct or indirectly, gave me the basic knowledge to carry out this work.

My accomplishments have not been mine alone, and over the years I have been encouraged, inspired, sustained and tolerated by some very important people in my life. I wouldn't have written these words if it wasn't for them. I'm dedicating this dissertation to all of them.

To my mother Helena and (late) father Carlos for their unconditional love and continued support throughout my whole life. They are the main reason I'm where I am right now.

To my best friend and brother Carlos Miguel for putting up with me and being there for me throughout not only this entire Master course, but also in everything else.

To my girlfriend Margarida for understanding and listening to my concerns and my outbursts, as well as her support and patience throughout this whole year.

To all my friends for their support and understanding, with particular appreciation to my friend Raquel, whose work ethic inspires me to be better every day.

It was a very intense year, full of challenges but it was much easier to overcome having great people like you, who supported and guided me through it all. Even though this work bears my name, it has a portion of all of you.

To all of you: Thank you!

Abstract

The quantity of biological data accessible in online repositories is exponentially growing, thus computer science methods are required for its management and analysis. Online, digital repositories are useful instruments for biodiversity research, as they provide fast access to data from different sources. Among these, large contributors are museums that hold a vast quantity of specimens in their collections.

This dissertation was developed as part of the COBIO-NET project, funded by FCT/AGA KHAN Foundation, and its goals were to create a comprehensive dataset on the marine coastal biodiversity of decapod crustaceans through different habitats in Mozambique and São Tomé and Príncipe, and to use digital tools to disseminate the compiled information online, so that its available to the scientific community and the general public. These digital tools have been used to aggregate, georeference and clean up global biodiversity data retrieved from online digital repositories, biodiversity data and scientific literature into a detailed dataset; to manage and structure biodiversity data for the compilation of biological studies through a relational database management system; to manage and process data through, also to show the acquired spatial data in the geographic information system QGIS and an interactive web map.

In this dissertation, a metadata structure was defined, which will be further used to store data collected during the COBIO-NET project. This metadata structure includes 26 fields defined by the Darwin Core metadata standard. A pilot dataset, based on this metadata structure was compiled, including 7486 decapod crustaceans occurrences records in Mozambique and São Tomé and Príncipe.

The database PostgreSQL constructed during this study is the digital support to manage and process information on decapod crustaceans from Mozambique and São Tomé and Príncipe. The data from the database can be filtered through queries, and this database is ready to be updated by researchers enrolled in COBIO-NET with more data from different marine invertebrate taxonomic groups.

The QGIS map design provides maps to visualize decapod crustacean occurrences, offering the foundation for the analysis of species incidence and allocation information. The map project is prepared in a way that it can be used to display information on larger datasets, including information on other marine taxonomic groups added during the COBIO-NET project.

A web map was constructed using Leaflet and is an interactive digital platform, that allows the visualization of decapod crustacean occurrences through mangroves, seagrasses, corals and other coastal areas of Mozambique and São Tomé and Príncipe. The web map provides distinct filters concentrations to manipulate the data, allowing visualization of occurrences events according with specific demands.

A dataset of biodiversity data, representing occurrence records, was compiled and the digital tools developed during this study will be published and disseminated through online digital repositories, that can be used in future studies to model crustacean species allocation studies. To evaluate species communities in an effort to estimate their probability of extinction and to preserve biodiversity. These data as well as the digital tools, and their following publication further adds and promotes our current knowledge on biodiversity of marine crustaceans in mangroves, seagrasses, corals and other coastal areas of Mozambique and São Tomé and Príncipe.

Keywords: dataset, Leaflet, PostgreSQL, QGIS, web map.

Resumo

As espécies costeiras e marinhas e os seus respectivos habitats estão a ser adversamente perdidos ou danificados, reduzindo significativamente a biodiversidade marinha. Os invertebrados marinhos, incluindo crustáceos decápodes, constituem importantes fontes de alimento para as populações locais, especialmente para pessoas mais pobres, que dependem destes recursos para sustento e alimentação. Moçambique e São Tomé e Príncipe abrigam um grande número de espécies decápodes indígenas, que são usadas pelas populações locais para a sua subsistência e segurança alimentar.

Existe uma grande quantidade de dados valiosos sobre a biodiversidade do mundo, armazenados em Coleções de História Natural, repositórios digitais, programas de vigilância climática, projetos de investigação e outros, que estão disponíveis para pesquisa. Em particular, as Coleções de História Natural albergam informação muito relevante para estudos de biodiversidade, incluindo séries espaciais e temporais, na medida em que são desenvolvidas desde o século XIX, e têm sido usadas em numerosos estudos, desde taxonómicos e sistemáticos até distribuição espacial e temporal e composição populacional.

A quantidade de dados acessíveis online cresce diariamente e desta forma, métodos e técnicas das ciências da computação são essenciais para a gestão e análise dos mesmos. Os repositórios online são instrumentos úteis na pesquisa sobre biodiversidade, pois fornecem acesso rápido aos dados de enumeras coleções. Vários museus, organizações e universidades processam informações das suas coleções em diversas bases de dados digitais que disponibilizam em repositórios na internet. Um desses repositórios de dados é o *Global Information Facility for Biodiversity* (GBIF). O modelo de metadados *Darwin Core (*DwC*)*, é usado pelo GBIF com o intuito de partilhar informação padronizada sobre biodiversidade, e desempenha um papel fundamental na interoperabilidade e integração destes dados.

Esta dissertação faz parte do projeto COBIO-NET – *Coastal biodiversity and food security in peri-urban Sub-Saharan Africa: assessment, capacity building and regional networking in contrasting Indian and Atlantic Ocean*, financiado pela fundação FCT/AGA KHAN. O objetivo desta seção do projeto COBIO-NET é compilar conjuntos de dados de crustáceos decápodes, e informação associada, de Moçambique e São Tomé e Príncipe por meio da catalogação em um conjunto de dados e base de dados organizados em DwC, e torná-los acessíveis através de repositórios e de mapas interativos online. Neste contexto, o principal objetivo da dissertação foi a construção de um conjunto de dados e o uso de ferramentas digitais para compilar informações a fim de criar um conjunto de dados abrangente sobre crustáceos decápodes e respetivos habitats nas zonas costeiras de Moçambique e São Tomé e Príncipe. O trabalho foi repartido em 4 partes: 1) recolha de dados mundiais sobre biodiversidade em repositórios digitais globais relacionados com Coleções de História Natural, outros dados de biodiversidade e literatura científica para a construção de um conjunto de dados; 2) gestão e processamento de dados sobre biodiversidade a partir da compilação de estudos biológicos por meio de um sistema de gestão de bases de dados relacional (SGBDR) de código aberto; 3) armazenamento e representação de eventos e distribuição de dados geográficos na ferramenta de um sistema de informação geográfica de código aberto; 4) disseminação de dados online através da criação de um *web map* interativo usando uma biblioteca JavaScript de *web mapping* de código aberto.

Nesta dissertação, a estrutura dos metadados é a base do projeto COBIO-NET, que será usada para armazenar os dados de invertebrados marinhos obtidos durante o projeto. O conjunto de dados piloto desenvolvido durante esta dissertação, apresenta um *layout* com 26 campos *Darwin Core*, e contém um conjunto de dados referente a 7486 ocorrências de crustáceos decápodes em mangais, pradarias marinhas, corais e outras áreas costeiras de Moçambique e São Tomé e Príncipe. Este modelo

de metadados é adequado para a recolha de informação pré-estruturada de acordo com o formato de dados usado pelo GBIF.

Ferramentas digitais tais como o *software Open Refine* e a linguagem de programação *Python* foram usadas para criar um conjunto de dados, através da compilação e limpeza de dados de ocorrências de biodiversidade obtidos a partir de repositórios digitais, dados obtidos diretamente de outras fontes e literatura científica, agregados em um conjunto de dados detalhado. De forma a poder realizar uma análise geográfica, é necessário que os dados incluam os locais de recolha na forma de coordenadas espaciais. Nos casos em que os registos obtidos não incluíam esta informação, a georreferenciação foi feita utilizando o software *GEOLocate*.

A base de dados *PostgreSQL* construída durante este estudo é o suporte digital usado para gerir e processar informações sobre crustáceos decápodes de Moçambique e São Tomé e Príncipe. Os dados da base de dados podem ser filtrados por meio de consultas e a base de dados pode ser atualizada pelos investigadores do COBIO-NET com mais dados referentes a outros grupos taxonómicos de invertebrados marinhos.

O *QGIS* foi o sistema de informação geográfica de código aberto utilizado para visualizar, processar e avaliar os dados e informações geográficas recolhidas. A avaliação geográfica, usando o *QGIS*, envolveu vários passos, como processamento de informações geográficas, mapeamento de diferentes habitats, combinação de camadas de informação e personalização da simbologia existente. O *design* do mapa *QGIS* foi projetado especificamente para o projeto COBIO-NET, oferecendo uma base para analisar informações sobre incidência e alocação de espécies. O projeto de mapeamento em *QGIS* permite a sua aplicação a outros conjuntos de dados mais amplos, incluindo informação sobre outros invertebrados marinhos.

O *plugin qgis2web* para o *QGIS* foi utilizado para produzir um mapa interativo em *JavaScript*, tornado público na internet, personalizado através da biblioteca *Leaflet* que permite visualizar as ocorrências das espécies de crustáceos decápodes. O mapa fornece vários filtros para manipular os dados, permitindo a visualização de ocorrências de acordo com critérios específicos. Este mapa inclui 18 camadas que podem ser escolhidas ou não para filtrar as informações visualizadas, e são categorizadas em (1) ocorrência de espécies, (2) habitat, (3) áreas dos países e (4) zonas costeiras globais.

Ao longo desta dissertação, foram usadas várias ferramentas e técnicas, que apresentaram vários desafios. A limpeza e validação de nomes científicos no conjunto de dados de biodiversidade foi um deles. O campo da taxonomia está em constante mudança, dificultando a compreensão de quais termos descritos correspondem ao nosso conhecimento contemporâneo de uma espécie em particular. As limitações também se estendem aos *shapefiles* de habitat usados neste estudo. A maioria dos conjuntos de dados possui cobertura global e foi compilada a partir de várias fontes de dados de qualidade e a escalas variadas para as quais a interpretação da imagem foi realizada. Outra limitação foi o mapa em *JavaScript* gerado com o *plugin qgis2web* no *QGIS*. Embora o *plugin* possa simular muitos elementos do *QGIS*, incluindo símbolos e estilos de camada, não consegue replicar aspetos mais complexos.

Os dados recolhidos no conjunto de dados podem ser partilhados através de repositórios online, a fim de serem usados, por exemplo, em estudos de distribuição de espécies e avaliar a composição de comunidades ecológicas distintas, para estimar a probabilidade de extinção e preservação da biodiversidade.

Outra aplicação deste trabalho é permitir a criação de uma coleção de dados de referência alargada a outros invertebrados marinhos de Moçambique e São Tomé e Príncipe, que seja

disponibilizada à comunidade científica e ao público em geral de forma a dar a conhecer a biodiversidade e história natural destes países africanos.

Uma das ideias do projeto COBIO-NET é gerir e centralizar todos os conjuntos de dados de biodiversidade produzidos durante o projeto e a documentação associada com recursos a ferramentas multimédia, num repositório digital. Este repositório será usado para disseminar as informações e dados reunidos para a comunidade científica e o público em geral. Portanto, o conjunto de dados, mapas *QGIS* e mapas interativos produzidos durante esta dissertação serão incluídos no repositório digital COBIO-NET para maior disseminação da informação a investigadores e sociedade em geral.

Além disso, as caraterísticas do conjunto de dados desenvolvido, que integra um conjunto de dados de biodiversidade de invertebrados marinhos estruturado em DwC, a base de dados, os mapas e o mapa *online* serão publicados em artigos científicos, *data papers* e repositórios digitais (GBIF e COBIO-NET) no âmbito do COBIO-NET, em colaboração com outros investigadores do projeto. Adicionalmente, ao serem disponibilizados ao público, estes dados e ferramentas digitais podem ser usados por outros utilizadores como ferramentas de ensino ou divulgação, para escrever livros, artigos científicos, folhetos de divulgação, etc.

Os crustáceos decápodes são elementos relevantes da dieta e meios de subsistência das populações locais de Moçambique e São Tomé e Príncipe. Neste contexto, os resultados obtidos nesta dissertação são relevantes na medida em que podem ser usados para ligar o conhecimento destes recursos naturais ao seu valor gastronómico nestes dois países, bem como aos Objetivos de Desenvolvimento Sustentável (ODS) das Nações Unidas, a saber, o ODS #2 (Erradicar a fome), que visa acabar com a fome, alcançar a segurança a alimentar e melhorar a nutrição; e o ODS #14 (Proteger a vida marinha), que se refere à biodiversidade marinha e costeira, sua conservação e uso sustentável para um desenvolvimento sustentável da sociedade humana.

Palavras-chave: *dataset*, *Leaflet*, *PostgreSQL*, *QGIS*, *web map*.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

%	Percentage
1:10m	1 centimeter on the screen is 100 kilometers in the real world
AToL	Assembling the Tree of Life
API	Application Programming Interface
CRS	Coordinate Reference System
CSV	Comma-separated values
DMS	Degrees, Minutes and Seconds
DOI	Digital Object Identifier
DwC	Darwin Core Metadata Model
EEZ	Exclusive Economic Zone
EML	Ecological Metadata Language
EoL	Encyclopedia of Life
EPSG	European Petroleum Survey Group
GBIF	Global Biodiversity Information Facility
GIS	Geographic Information System
gni	Global Names Index
GPS	Global Positioning System
GREL	General Refine Expression Language
HTML	Hypertext Markup Language
iDigBio	Integrated Digitized Biocollections
IIP	National Institute of Fisheries Research
INBO	Institute for Nature and Forest
ITIS	Integrated Taxonomic Information System
MNHN	Muséum national d'Histoire naturelle

MOZ	Mozambique
MSSQL	Microsoft SQL Server
MUHNAC	Museu Nacional de História Natural e da Ciência de Lisboa
NCBI	National Center for Biotechnology Information
NHC	Natural History Collection
NHMUK	Natural History Museum, London
NRM	Swedish Museum of Natural History
OBIS	Ocean Biogeographic Information System
ODBC	Open Database Connectivity
OGC	Open Geospatial Consortium
RDBMS	Relational Database Management System
RMCA	Royal Museum for Central Africa
SAM	Iziko South African Museum
SAMC	Iziko South African Museum
SDG	Sustainable Development Goals
SQL	Structured Query Language
STP	São Tomé and Príncipe
SVG	Scalable Vector Graphics
TDWG	Biodiversity Information Standards
UN	United Nations
URL	Uniform Resource Locator
USNM	Smithsonian National Museum of Natural History
WCS	Web Coverage Service
WFS	Web Feature Service
WFS-T	Transactional Web Feature Service

WGS	World Geodetic System
WMS	Web Map Service
WoRMS	World Register of Marine Species
ZMC	Zoologisk Museum Copenhagen
ZMH	Zoological Museum Hamburg
ZMMU	Zoological Museum of the Moscow State University

1. Introduction

The growth of human populations within coastal areas has increased due to rural-urban migration, with people relocating to more urbanized and economic centers. This migration increases human pressure on the environment due to land and marine-based human activities. Nearshore habitats are of great socio-economic significance, especially in the western Indian Ocean, since 65 million people live within 10 km of the coast in the greater Indian Ocean region (Burke *et al.,* 2011). As a result, coastal and marine living resources and their habitats are being adversely lost or damaged reducing marine biodiversity. Marine invertebrates, including decapod crustaceans, comprise important food sources for local populations, especially for the poorest people, that depend on these resources for their livelihoods and food security.

Among crustaceans (Phylum Arthropoda, Subphylum Crustacea), the order Decapoda is the largest with nearly 15,000 extant species and approximately 3.000 genera (De Grave *et al.*, 2009). The decapod crustaceans include diverse living species that inhabit a wide variety of habitats (Mantelatto and Sousa, 2000). Mozambique (MOZ) and São Tomé and Príncipe (STP), respectively on the eastern and weastern coastlines of Africa, are home to a large number of indigenous decapod species. The species present in both countries belong to a significant part of all classified families of decapods, with over 100 families present of 180 classified families (De Grave *et al.*, 2009; Emmerson, 2016). Brachyuran crabs alone have over 7,000 extant species and over 400 species have been identified as inhabiting the Western Indian Ocean (WIO) (Ng *et al.*, 2008). Decapods are of high commercial, gastronomic and ecological importance (Silva *et al.*, 2004; Almeida *et al.*, 2008). For instance, a third of MOZ's population lives along the coast and depends on resources in those areas to sustain their livelihoods (Francis and Torell, 2004). Decapods are used as food source all over the world and are one of Mozambique's main fishery products. Deep and shallow water species are targeted by fleets which work in three sectors: 1) artisanal, 2) semi-industrial and 3) industrial. Artisanal catches from mangroves are of high importance since most coastal adult population is employed in fisheries (Hutchinson *et al.*, 2014). The fisheries sectors in STP, on the other hand, are shaped mainly by traditional, small-scale fishing activities, built exclusively by the resources available in the country national waters, particularly those closest to shorelines. A situation that is common to most African fisheries, because of insufficient capital for investments in the fishery sector (Heck *et al.*, 2007).

Digital repositories are valuable tools in biodiversity studies, since they provide centralization of the available global information, allow prompt accessibility, integrate data from various sources around the world enabling more holistic data analysis and reliable reproducible studies (Maldonado *et al.*, 2015), and they also reduce financial and biodiversity costs of more *in situ* collecting events. Biodiversity repositories have been developed since the 1800's as natural history collections (NHC) which have been used in diverse studies from taxonomic and systematic information, geographic and temporal distributions, and population composition (Suarez and Tsutsui, 2004). NHC convey important scientific infrastructures of valuable data regarding the world's biodiversity because they comprise organized sets of natural objects collected over time, in various locations, with relevant information associated (Cartaxana *et al.*, 2014). These collections play an indispensable and often overlooked role in our current knowledge, as they do not only open an important "window" to the current and past diversity of life on Earth, but also play a vital role in fostering cutting-edge research in many areas (Lane, 1996). Hence, NHC contain valuable data and specimens used in research, higher education and science communication. NHC are a result of centuries of natural history and systematic studies, environmental monitoring programs, sound archives, research projects, among others, which have collected and stored data on the world biodiversity. All the collected information is stored and managed in many different systems and environments, that vary widely depending on the type of details captured

and stored for any individual record (Schindel and Cook, 2018). However, the standardization and availability of information from NHC on the internet has been increasing, allowing rapid access to data from various collections, as well as a greater taxonomic efficiency and quality control in the use of scientific nomenclature. Several museums, institutions and universities are digitizing and processing data from their collections, making it available on online repositories, which hold information on diversity, taxonomy and historical distributions of species (Graham *et al.*, 2004). One of these repositories is the Global Biodiversity Information Facility (GBIF), which supports the publication of datasets using widely accepted biodiversity data standards. Currently, the GBIF only publishes datasets directly from organizations, therefore, any individual wishing to publish relevant data sets should work through their affiliate organizations or consider submitting the data to a journal (Robertson *et al.*, 2014).

Other online repositories such as Integrated Digitized Biocollections (iDigBio), VertNet, World Register of Marine Species (WoRMS), Integrated Taxonomic Information System (ITIS) and citizens-based like iNaturalist and Biodiversity4all, are available, and often complement each other. Other sources of biodiversity data include academic publications and checklists, either on digital (open-access or not, with or without specific Digital Object Identifiers (DOI)) or in paper support that are often more specific and scattered. Therefore, gathering data included in these sources on more broad datasets, is also an important step to make the information available to different purposes and to wide audiences.

Given the increasing rates of biodiversity and habitat loss, it becomes imperative that we establish a simpler and more efficient form to integrate all biodiversity information in datasets and promote open sharing of associated data, so that scientists, analysts and politicians can both understand and apply them in research and policy decisions (Rocha *et al.*, 2014). Marine biodiversity databases are important tools, which allow research, capacity-building and society outreach. Exploring existing digital records of decapod crustaceans biodiversity from MOZ and STP, including global digital repositories, NHC records and scientific literature; compile them into a comprehensive dataset; generate species occurrences distribution maps; and make these available online through interactive platforms; will increase our understanding of decapod crustaceans occurrences and distribution along the coast of both countries and will contribute to science, policy making and legislating, as well as to engage Mozambican and São Tomé and Principe's citizens with science and the preservation of their natural resources.

1.1. Biodiversity Dataset Construction

Datasets comprise a collection of data organized into rows and columns where each column represents a particular variable and each row corresponds to a given record of the dataset. These data are prepared for statistical analysis. The dataset structure and properties are defined by several characteristics, that should be standardized in accordance to the purpose of the dataset. Despite the increasing number of technological tools available, a great deal of work is still being done to integrate biodiversity information on online digital repositories. For example, it is estimated that only 5% of NHC data is captured in digital databases (Beaman and Conn, 2003). The Darwin Core (DwC) metadata model defines standards for a straightforward and flexible framework for compiling biodiversity data from varied and variable sources, ensuring data interoperability. Originally developed by the Biodiversity Information Standards (TDWG) community, DwC is maintained by the evolving community (Wieczorek *et al.*, 2012). It plays a key role in sharing, using and reusing open access biodiversity data and the vast majority of the hundreds of millions of species records available through GBIF are currently in this format (Baskauf *et al.*, 2016). In fact, the DwC serves to share biodiversity data and first emerged in 1999 as a loosely defined set of terms. It has developed through several iterations across different groups, resulting in many different variants. A formal set of terms and processes for managing change was needed to ensure the usefulness of data integration. The objective has been to keep DwC as simple and open as possible, developing terms only when there is a shared

demand for them. DwC has a relatively long history of community development and is widely used (Wieczorek *et al.*, 2012).

Once the data is incorporated into a dataset, additional work is generally required to make the sample record usable for further analysis. To do that, data must be validated and follow certain data quality requirements. Some common errors that hamper data quality are summarized in Table 1.1.

Table 1.1 *– Type of errors that can be found in data quality assessments (Chapman, 2005).*

Data values	**Description**
Duplicated	Two or more occurrences records referring to exactly the same occurrence (e.g. records from two or more different sources).
Inconsistent	Same records from multiple sources with different formats, languages or other criteria (e.g. same institution, different acronyms).
Incorrect	Values that do not match a given field within its context and possible values (e.g. misspellings of localities).
Missing	Non-existent values where data should occur (e.g. if a lower taxonomic rank is included, all higher ranks should be included as well).
Non-atomic	Values lumped together into a single cell that should be split in two or more fields (e.g. latitude and longitude coordinates in the same cell).
Unintended	Using dubious values in species (e.g. 'to be determined', 'sp. nov.').

Verification and validation of data is a very important step in correcting and homogenizing data, according to a defined metadata model. For example, the dates may be in different formats (01-02-1990; 1990-02-01; 02-01-1990; February 1, 1990, etc.). This information will need to be standardized. There are several tools available for manipulating and correcting large datasets such as the spreadsheet software, Open Refine, and the programming languages R and Python. More complete, accurate and organized data have greater scientific value, and are easier to work with and publish on local or international repositories, in data papers, etc.

The spatial coordinates of the sampling, or observation locality, are extremely crucial information for biodiversity records. The term spatial data denotes any type of data, that describes phenomenon associated with some spatial dimension. Georeferenced data belong to a particular class of spatial data, which describe events, objects, and phenomena in the world associated with their location on the surface, at a certain moment or period of time (Câmara *et al.*, 1996). While advances in technology have made portable Global Positioning System (GPS) receivers a practical solution for capturing spatial coordinates as they are gathered, text-based location descriptions remain essential and sometimes the only information available. The process of converting a text-based description based on the locality of the record into geospatial coordinates is called geocoding (Murphey *et al.*, 2004, Wieczorek *et al.*, 2004) or retrospective georeferencing, if added later by others than the collector (Chapman, 1991). Geocoding describes the process of locating an entity in real-world coordinates. It has become more prominent in recent years, due to the increasing availability of Geographic Information System (GIS) tools to facilitate this process. The geocoding process is, ultimately, a set of methods for overcoming problems associated with misinterpreted localities in museum specimens in NHCs. In fact, museum curators and technicians, currently face the great challenge of assigning spatial coordinates to locality data, for hundreds of millions of antique species records in NHC (Baker *et al.*, 1998). The base information from older collections often lacks

geospatial coordinates, needed to track biodiversity change over time. Unfortunately, the older the collection data, the less accurate it usually is. This is partly due to the changing geographical terminology and collection standards over time (Chapman and Milne, 1998). Another significant factor is the disassociation of detailed records of collecting events (text, maps, photographs, coordinates), of sample labels and collection catalogs (Bloom *et al.*, 2018). Nowadays, museums are starting this process with the common goal of making data available for geospatial analysis and their quality directly determines their usability of the data in geospatial analysis (Brisby, 2000). This relatively new activity is leading to the development of methodologies and tools to streamline the process, through a combination of shared and independent methods. Several projects have developed guidelines, protocols and tools for geocoding biodiversity information such as BioGeomancer (Chapman and Wieczorek, 2006), GEOLocate (Rios and Bart, 2010), MaNIS (Wieczorek *et al.*, 2004; Wieczorek and Wieczorek, 2015) and MapSteDI (Murphey *et al.*, 2004).

However, if the goal is to achieve comparable georeferenced data, developing different standards works against it. Georeferencing refers not only to assigning a point to a location, but also to determining the certainty of the assigned point. If properly implemented, georeferencing increases the accuracy of raw locality data, while turning it into an analytically useful format (Marinoni and Peixoto, 2010). Since, the degree of accuracy and precision processed by the georeferencing procedure depends on raw locality data, the geocoding process should include the attribution of some measure of degree of error or reliability in the data. For the most part, methods that currently include such confidence measures are expressed and derived in different ways. Perhaps the most important aspect of georeferencing is that it is a reproducible process. This is largely responsible for the impetus behind the georeferencing efforts, because it means that the process can transform location data consistently and accurately. The importance of these efforts is emphasized by the underlying purpose: to greatly increase data available for biological distribution and diversity studies. The ultimate goal is a quantitative and qualitative biological history derived from geospatial and temporal information (Matos, 2013).

1.2. Data Management

Data can be stored and maintained in databases in order to share and exchange data. Databases hold information in a manner that ensures the efficient management of large quantities of data and the retrieval of information as required to perform analysis and are intended for data research and interoperability with other databases (Shanmughavel, 2007).

The PostgreSQL is a POSTGRES Version 4.2 based relational database management system (RDBMS), developed by the Department of Computer Science at the University of California, Berkeley. POSTGRES pioneered a number of concepts that became available much later in some commercial database systems. PostgreSQL is an open-source descendant of this original Berkeley code, which supports much of the Structured Query Language (SQL) standard and offers many modern features. PostgreSQL may be used, modified and distributed by anyone for any purpose, whether private, commercial or academic, free of charge (The PostgreSQL Global Development Group, 2019). Using a RDBMS like PostgreSQL can be advantageous, since it allows the integration of standardized biodiversity data from datasets, manage data access to multi-users, and its compatibility and facility in management geographic data by integrating the database into GIS tools (Senterre and Wagner, 2016).

1.3. Geographic Data Representation

An important tool for biodiversity data management is GIS, which can accommodate and display various sources for spatial information and analysis (Wahid, 2006). Several researchers have studied distribution patterns in marine species, giving every information regarding population regulation, local depletion, migration, marine reserve design and habitat selection, amongst others (Freire and Gonzalez-Gurriaran, 1998; Hooge and Taggart, 1998). However, little of the distribution pattern analysis developed and used in the terrestrial environment has been applied to aquatic species. In fact, despite the extensive development and use of analytical methods in spatial patterns studies on terrestrial species, using these techniques with GIS in aquatic systems have been difficult due to the lack of specific and suitable tools (White and Garrott, 1990; Hooge *et al.*, 1999). Nevertheless, the combined use of radio-tracking, satellites, GPS and GIS systems, are becoming more used by scientists in aquatic studies. Moreover, the increase of georeferenced species occurrences data enables the use of GIS tools that can be used for geographic data representations through, e.g., the creation of accurate distribution maps.

The advantage of a GIS is that it modulates reality based on data and plays a major role in today's society, as it is an information system designed to capture, model, store, receive, share, manipulate, analyze and present geographically referenced information (Worboys and Duckham, 2004). Basic GIS operations now provide a secure basis for measuring, mapping and analyzing data. Data stored in a GIS database provides a simplified version of the Earth's surface. Georeferenced data can be organized by a GIS using different criteria, e.g. thematic maps or spatial objects. Each thematic layer can be saved using an appropriate data format, depending on its nature and the purpose of its use (Santos, 2013).

For example, in a study by Ramos *et al.* (2012), location descriptions were initially cleaned in GIS using administrative maps. The result was a georeferenced database with 2067 occurrences records of 47 nationally threatened forest species. Each record had a unique point resource by species and sufficient metadata directing the database user to the hit data source. The database can be used as a tool in determining priority species for species, taxonomic identification and historical mapping. It also serves as an integral component in the spatial modeling of tree species distribution and forest formations in the past and in a possible future scenario.

QGIS is a free and open-source geographic information application system that allows the visualization, editing, and analysis of spatial data. QGIS provides a rich environment for analyzing motion patterns, it allows the integration of multiple layers of habitat data into a structure capable of complex two-dimensional and three-dimensional analysis. QGIS integrates other open-source GIS packages, including PostGIS, GRASS and MapServer, to provide users with extensive functionality (Raes and van't Zelfde, 2016). One of the essential aspects of QGIS is that it works with geographic coordinates in the decimal format (e.g. 37.38º N).

1.4. Online Data Dissemination

Web mapping is the process of using GIS maps to interactively navigate and are accessed through a web browser. Since their conception, web maps have become more important than the historical and static cartography. This is because, from the moment data and geographic information are available on the web, the user chooses what to see, interacting with the map in search of understanding the various possibilities, depending on the exposed layers. The use of interactive platforms comprising web maps are becoming more important not only for the scientific community, but also for the civil society and decision-makers as they allow the

presentation of data in a simple and understandable visual scheme, as shown in the studies of Cristofori *et al.* (2015, 2016b and 2017) and Vincent *et al.* (2018).

Making web maps available online to the public can be accomplished using, for example, the QGIS Cloud online, which allows the creation of maps and can then be published and mapped with the QGIS Cloud plugin. QGIS Cloud data management offers a PostgreSQL 9 database with the PostGIS spatial component. New databases can be created and managed with any compatible administration tool, such as pgAdmin. The data and maps can be shared through Open Geospatial Consortium (OGC) compliant web services (e.g. Geoserver) via the three web service standards: 1) Web Map Service (WMS) to display maps, 2) data downloaded via Web Feature Services (WFS) and 3) Web Coverage Service (WCS). These allow web clients to query and receive geographic information in the form of image, vector, or coverage data. With the transactional Web Feature Service (WFS-T), spatial data can be directly edited through the web service (Sourcepole AG, 2018). Another option for publishing web interactive maps from QGIS, is OpenLayers, one of the most popular web publishing tools, which is an open-source JavaScript library. OpenLayers is modular, high-performance, and has many functions for displaying and interacting with maps and geospatial data (Henriques, 2016). Finally, Leaflet is a JavaScript library that incorporates maps, while more recent than OpenLayers, it quickly gained popularity as it works efficiently on all major desktop and portable platforms out of the box, leveraging HTML5 and CSS3 in modern browsers. It can be extended with add-ons, has an easy-to-use, well-documented Application Programming Interface (API), and a simple, readable source code (Agafonkin, 2019).

1.5. Objectives

This dissertation is integrated in the COBIO-NET project: Coastal biodiversity and food security in peri-urban Sub-Saharan Africa: assessment, capacity building and regional networking in contrasting Indian and Atlantic Ocean. Accordingly, the current work was developed with specific objectives within the scope of COBIO-NET.

The main objective of this dissertation was to integrate comprehensive information on decapod crustaceans inhabiting mangroves, seagrasses, corals and other coastal areas of MOZ and STP, and to disseminate this information online by constructing a dataset and making it available through an interactive digital platform where this data can be accessed, filtered and mapped. Understanding the spatial occurrences and distributions of decapod crustaceans in MOZ and STP will provide an important baseline information, trends on costal marine biodiversity and food security in these two countries. Furthermore, this work aimed to structure a dataset ready to be updated with new marine invertebrate data and implement digital tools to link these data, allowing the generation of online maps for interactive visualization.

Therefore, the following specific objectives were addressed:

- Dataset construction: aggregate global biodiversity data on decapod crustaceans from MOZ and STP, scattered in digital repositories (and to some extent on non-digital archives) relating to biodiversity data sources such as NHC and scientific literature, into a detailed dataset to create a reference biodiversity collection, expandable by other marine invertebrate data compiled by the COBIO-NET researchers with the future purpose to make all the information publicly available and accessible through the publication of the entire dataset on global biodiversity repositories (e.g. GBIF).

- Data management: develop a data management structure to manage and process biodiversity data from the previous dataset through an open-source relational database management system (RDBMS).
- Geographic data representation: construct maps with geographic data to visualize and analyze occurrence and distribution data of decapod crustaceans, included in the dataset using an open-source geographic information system (GIS) application tool.
- Online data dissemination: develop of an interactive digital platform, based on the dataset information, to visualize decapod crustaceans occurrences and distribution through different habitats using an open-source JavaScript library designed for web mapping and make it accessible to the scientific community and the general public.
- Apply the dataset and platform to address specific issues, namely species occurrences and distribution of decapod crustaceans from mangroves, seagrasses, corals and other coastal areas along the coastlines of MOZ and STP.

1.6. Dissertation Online

This outline gives an overview of the main points of the dissertation and it clarifies the framework and the focus of the dissertation.

Table 1.2 – *Schematic outline of the dissertation.*

Methodology	**Results**	**Discussion**
⇓	⇓	⇓
Biodiversity Dataset Construction Data structuring Data compilation Data geocoding Data quality Coding	Biodiversity Dataset Construction	Dataset Construction: development of a Dataset of MOZ and STP decapod crustaceans Limitations on the development of the dataset Different approaches to the used methodology
↓	↓	↓
Data Management PostgreSQL	Data Management	Data Management: using the PostgreSQL RDBMS Limitations on data management Different approaches to the used methodology
↓	↓	↓
Geographic Data Representation QGIS	Geographic Data Representation	Geographic Data Representation: mapping with QGIS Limitations to the shapefiles data Different approaches to the used methodology
↓	↓	↓
Online Data Dissemination Leaflet	Online Data Dissemination Species occurrences Habitats Country boundaries Appearance Online access	Online Data Dissemination: web maps for geographic visualization Limitations to the online data dissemination Different approaches to the used methodology
↓		↓
Data Analysis		Applications of the Developed Digital tools
		↓
		Long-Term Goals and Perspectives

2. Methodology

This study comprised the employment of digital tools, see Figure 2.1, applied for data collection, to construct a comprehensive dataset on decapod crustaceans and respective habitats through coastal zones in MOZ and STP, data processing and management, geographical data representation, and online data dissemination through an interactive digital platform, as follows:

- Dataset Construction: aggregation of global biodiversity data contained in digital repositories relating to NHC, biodiversity data and scientific literature into a detailed dataset, using coding of different Python scripts to automate processes within the application.
- Data Management: management and processing of biodiversity data from biological research collections through an open-source relational database management system (RDBMS).
- Geographic Data Representation: manipulation and presentation of occurrence and distribution spatial data in an open-source geographic information system (GIS) application tool.
- Online Data Dissemination: development of an interactive digital platform map using an open-source JavaScript library designed for web mapping.

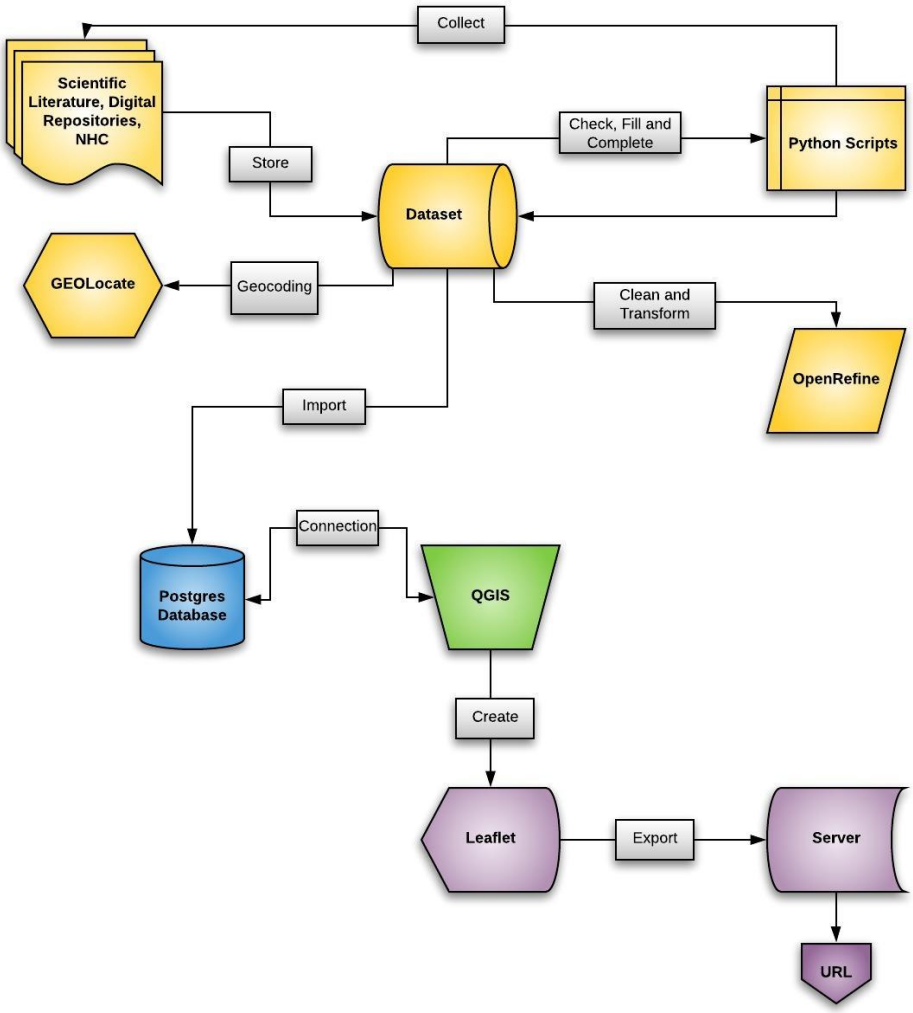

***Figure 2.1** – Workflow of the study project, depicting the main tasks of 1) data collection in yellow, 2) data management in blue, 3) data representation in green and 4) data dissemination in purple (created using the Lucidchart web-based application).*

2.1. Biodiversity Dataset Construction

The construction of a dataset involved the compilation and organization of decapod crustacean data from STP and MOZ contained in digital repositories relating to NHC, biodiversity data and scientific literature. The dataset was compiled and organized in a Microsoft Excel spreadsheet, in which each row corresponds to a single occurrence in the dataset, i.e. an observation or sample, in a defined geographic location and period of time. Only the occurrences with taxon rank equal to genus or more specific, such as species and subspecies, were considered for this dataset. The data was catalogued into a DwC structured dataset, compiled, geocoded and validated.

2.1.1. Data Structuring

The majority of the attributes included in the dataset were defined in accordance to the corresponding DwC column terms (TDWG, 2014):

- **catalogNumber**, unique identifier for the record within the data set or collection.
- **scientificName**, full scientific name, with information on author and date, if available. When forming part of an Identification, this should be the name in the lowest level of taxonomic rank that can be determined.
- **kingdom**, full scientific name of the kingdom in which the taxon is classified.
- **phylum**, full scientific name of the phylum or division in which the taxon is classified.
- **class,** full scientific name of the class in which the taxon is classified.
- **order,** full scientific name of the order in which the taxon is classified.
- **family,** full scientific name of the family in which the taxon is classified.
- **genus,** full scientific name of the genus in which the taxon is classified.
- **infraspecificepithet,** name of the lowest or terminal infraspecific epithet of the scientificName, excluding any rank designation.
- **taxonRank,** taxonomic rank of the most specific name in the scientificName.
- **country,** name of the country or major administrative unit in which the location occurs, using the controlled vocabulary Getty Thesaurus of Geographic Names.
- **locality,** specific description of the place. This term may contain interpreted information modified from the original to correct perceived errors or standardize the description.
- **verbatimLocality**, original textual description of the place.
- **habitat**, category or description of the habitat in which the event occurred (characterized by Python code. See Section 2.1.5.).
- **decimalLatitute,** geographic latitude (in decimal degrees, using the spatial reference system given in geodeticDatum) of the geographic center of a location. Positive values are north of the Equator, negative values are south of it. Legal values lie between -90 and 90, inclusive.
- **decimalLongitude,** geographic longitude (in decimal degrees, using the spatial reference system given in geodeticDatum) of the geographic center of a location. Positive values are east of the Greenwich Meridian, negative values are west of it. Legal values lie between -180 and 180, inclusive.

- **geodeticDatum,** ellipsoid, geodetic datum, or spatial reference system (SRS) upon which the geographic coordinates given in decimalLatitude and decimalLongitude are based.
- **individualCount,** number of individuals represented present at the time of the occurrence.
- **eventDate,** date-time or interval during which an event occurred. For occurrences, this is the date-time when the event was recorded. Not suitable for a time in a geological context.
- **institutionCode,** name (or acronym) in use by the institution having custody of the object(s) or information referred to in the record.
- **occurrenceID,** identifier for the occurrence (as opposed to a particular digital record of the occurrence). In the absence of a persistent global unique identifier, construct one from a combination of identifiers in the record that will most closely make the occurrenceID globally unique.
- **references,** resource that is referenced, cited, or otherwise pointed to by the described resource.

According to the specificity of the data been studied, a few new attributes not present in DwC were created in the dataset:

- **species,** full scientific name of the species in which the taxon is classified, concatenation of genus and specificEpithet (name of the species epithet of the scientificName).
- **previousName,** previous full scientific name in which the taxon was classified (checked by Python code. See Section 2.1.5.).
- **depthInMeters,** the depth below the local surface.
- **taxonInfo,** external link of information to the website WoRMS (filled by Python code. See Section 2.1.5.).

2.1.2. Data Compilation

The data was gathered from worldwide, open-source, available information from a) NHC records, b) online digital repositories and c) scientific literature. The following sources were used for each step:

a) NHC records

Museums such as the Smithsonian National Museum of Natural History (USNM), Naturalis Biodiversity Center, Muséum national d'Histoire naturelle (MNHN) and Royal Museum for Central Africa (RMCA), which have their collections information available online. The download was accomplished directly from all museums' websites mentioned, with the exception of the MNHN. The data was instead retrieved through a Python script written specially for this purpose (See Section 2.1.5.). For such cases, in which NHC records were not available online, data was requested directly to the respective curator, e.g. Museu Nacional de História Natural e da Ciência de Lisboa (MUHNAC) and Swedish Museum of Natural History (NRM). The data was retrieved in the last quarter of 2018. Sources used are listed in Section 6.1.

b) Online Digital Repositories

– GBIF (Global Biodiversity Information Facility)

At gbif.org, the search for occurrences matching the desired criteria was conducted through "Occurrences" search under the "Get data" submenu at https://www.gbif.org/occurrence/search. Filtered by specific fields such as "Scientific name" and "Country or area", with the terms "Decapoda", and "Mozambique" and "Sao Tome and Principe", respectively (see Figure 2.2). The data was downloaded in Comma-separated values (CSV) format. Retrieved on 21 September 2018. Sources used are listed in Section 6.1.

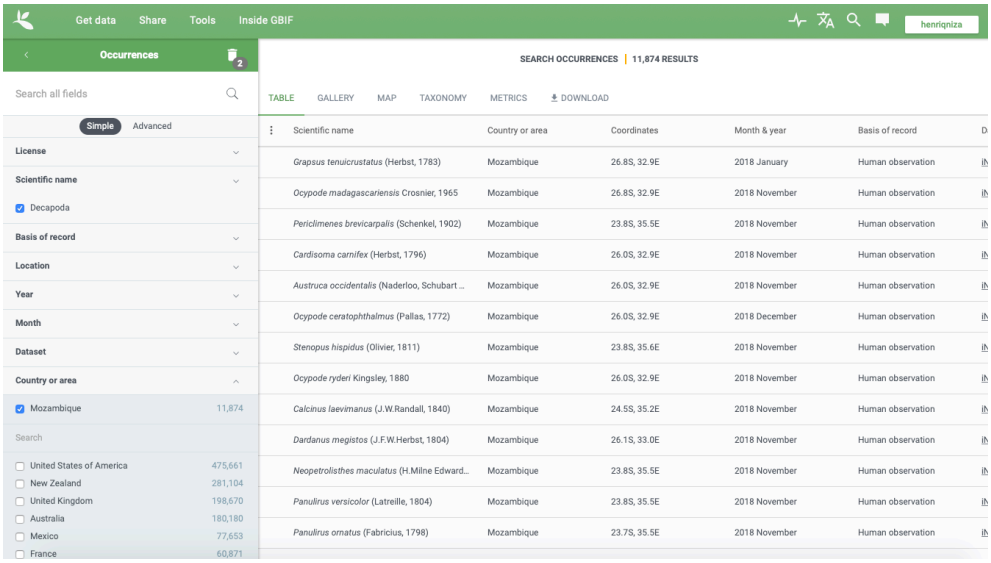

***Figure 2.2** – GBIF website with a descriptive search conducted under Get data, Occurrences submenu (GBIF.org, 2018).*

– iDigBio (Integrated Digitized Biocollections)

At idigbio.com, inside the portal, the search was filtered by "Order" and "Country", with the terms "Decapoda", and "Mozambique" and "Sao Tome and Principe", respectively. The data was downloaded in CSV format. Retrieved on 7 November 2018. Sources used are listed in Section 6.1.

– OBIS, former IOBIS (Ocean Biogeographic Information System)

At iobis.org, now obis.org, the data was filtered using the search terms, "Decapoda", "Mozambique" and "Sao Tome and Principe". The data was downloaded in XLSX format. Retrieved on 3 November 2018. Sources used are listed in Section 6.1.

c) Scientific Literature

Species checklists were used to better understand existing species of decapod crustacean from MOZ and STP (e.g. Bandeira, S. & Paula, J. (eds.) (2014). *The Maputo Bay Ecosystem.* WIOMSA, Zanzibar Town, 427; De Grave & Fransen (2011). *Carideorum catalogus (Crustacea: Decapoda).* Zool. Med. Leiden 85; Bento, M. & Paula, J. (2018). *WIO Journal of Marine Science* 17, 13-51; Charles L Griffiths (2017). *A Guide to, and Checklist for, the Decapoda of Namibia, South Africa and Mozambique.* African Zoology 52, 73). The

search of scientific literature was based mainly in the freely accessible web search engine Google Scholar, using keywords such as e.g. “decapods”, “Decapoda”, “Maputo”, “Inhaca”, “Mozambique”, “São Tomé and Príncipe”. Other sources used were the websites National Center for Biotechnology Information (NCBI) database PubMed at https://www.ncbi.nlm.nih.gov/pubmed/, ScienceDirect at https://www.sciencedirect.com, ResearchGate at https://www.researchgate.net and Assembling the Tree of Life (AToL): Decapoda at https://decapoda.nhm.org. The data was retrieved in the last quarter of 2018. Sources used are listed in Section 6.1.

2.1.3. Data Geocoding

Geocoding is the process of taking coded location information, such as addresses or grids and turning it into explicit location information – X and Y coordinates. Geocoding biodiversity data records is imperative so it can be related to a ground system of geographic coordinates. However, a large percentage of the data collected did not have latitude/longitude coordinates. Therefore, geocoding, verification and correction of geographic coordinates was accomplished using the GEOLocate Collaborative Georeferencing Web Client interface at https://www.geo-locate.org/web/default.html. GEOLocate georeferencing services were introduced in 2005 and these services allow software developers, to integrate georeferencing directly into their databases and applications. Specifically, the application enables collection management by providing users with direct access to geo-referencing, without having to export and import data or run additional software. A number of web-based clients also use GEOLocate e.g. Arctos, Specify, Symbiota and Tropicos. (Geo-locate.org, 2019).

GEOLocate uses an algorithm that begins by standardizing the locality sequence, in common terms, analyzes distances, compasses directions, and important geographic identifiers. This information is then used in a series of surveys and displacement calculations, to determine geographic coordinates. The resulting coordinates are then sorted, based on the type of information on the digital map display, for user verification, correction and error determination. In addition to automated location description processing, one of GEOLocate's goals is to provide an interface through which users can geo-reference records one by one in batches of files, view and correct calculated coordinates and determine polygonal error descriptions. After the coordinates derived from a location, then description and adjustments can be made by clicking and dragging a point displayed on a map. Error estimates can subsequently be recorded, as the maximum length a description can occupy. GEOLocate also allows the creation of shared communities, in which the users can co-operatively review and edit, by providing comments and saving them to the shared dataset in the collaborative georeferencing framework. See Figure 2.3.

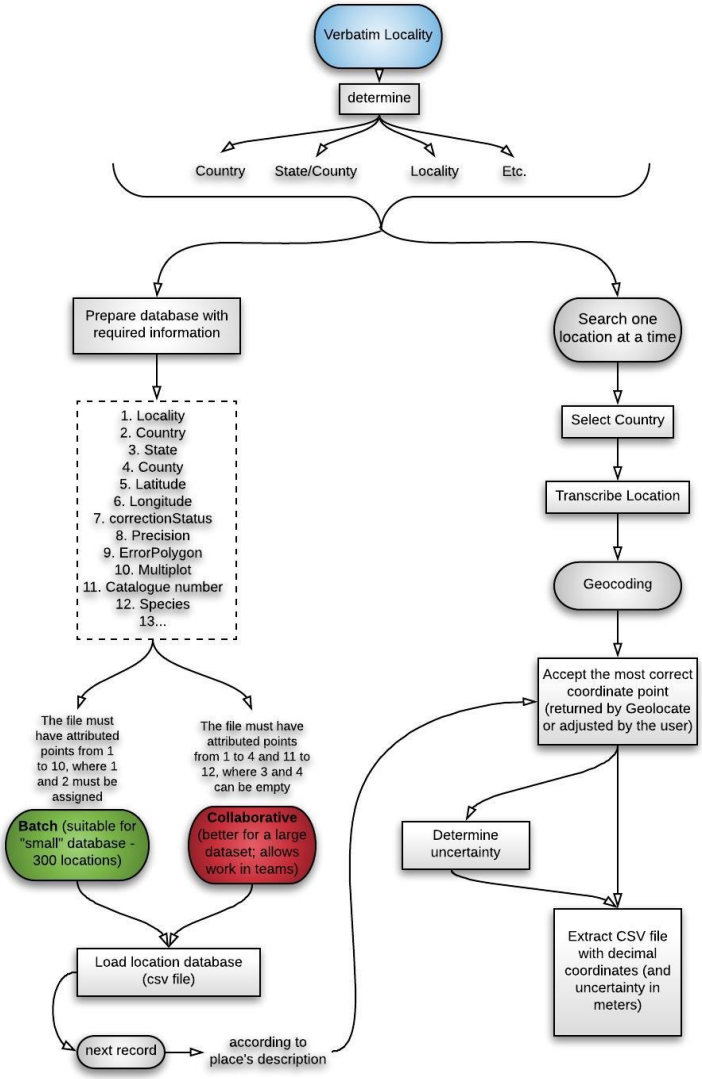

***Figure 2.3** – Workflow of how to geocode data in GEOLocate directly from the web browser, using available web-based clients (Adapted from Soares, 2019; created using the Lucidchart web-based application).*

GEOLocate tasks included a) uploading data and b) geocoding data associated with locations.

a) Uploading Data

To upload the data into the GEOLocate Collaborative Georeferencing Web Client interface, data from the two countries studied, MOZ and STP were split into two distinct datasets. Data was further split into different files according to the verbatimLocality language: English, Spanish, French, German, Dutch and Portuguese. Entries with sourced coordinates were excluded from the analysis.

The collaborative georeferencing framework was used in the analysis, as it is ideal for largescale multi-institution projects. The data source fields catalogNumber, scientificName, country and verbatimLocality were

mandatorily used for the analysis and were mapped into the desired system field for uploading. See Figure 2.4.

Figure 2.4 – *Web-based data management portal for mapping fields in data source. Required fields are indicated in dark grey. Matching field are mapped with auto-mapping (coge.geo-locate.org, 2019).*

b) Geocoding Data

To geocoding the occurrences, it was necessary to search one location at a time. According to the local description, the coordinates' point that were most correct were accepted, either determined automatically by the GEOLocate or manually by the user.

Most of the records were geocoding using the "Place marker" automatic option, via a similarity index which is used to identify all records that appear to describe the same collection locality, regardless of syntax.

Records such as e.g. "200 m west of Lagoa Azul; São Tomé e Príncipe" were also geocoded by using the "Measure" option to re-classify records that had to be manually corrected. See Figure 2.5.

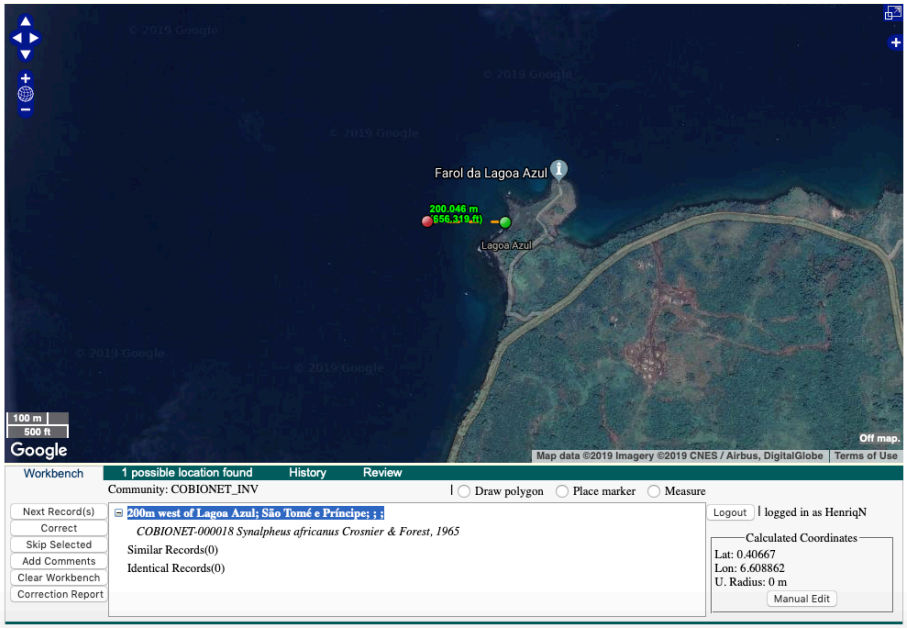

Figure 2.5 – *Correction depicted with the "Measure" tool. The location of the green marker was dragged and dropped in the desired location. The "Correct" button saves the corrections and moves to the next uncorrected locality (geo-locate.org, 2019).*

Records with the general description of "Off" e.g. Zambeze River, Off Mouth, were geocoded 200 to 300 meters from the locality. The uncertainty of the records was dismissed. The georeferenced work was extracted in a CSV file. Any records that were unable to be geocoded by GEOLocate, were either discarded or manually searched and georeferenced using Google Maps.

2.1.4. Data Quality

The data saved in a spreadsheet was collected from multiple sources. Therefore, ensuring data standardization was essential to produce a reliable dataset. To obtain such a dataset, the data was a) cleaned, b) transformed and c) extended by using relational information. In order to achieve this, the open-source desktop application OpenRefine v3.1 was used to wrangle the data.

a) Data Cleaning

The various sources were compared amongst each other in order to spot duplicates, erroneous data and when possible to fill in missing information.

Using the "Text facet" feature for text columns, cell contents were grouped into unique entries allowing the quick visualization of similar data written in different ways. For example, in Figure 2.6., one can observe how "Beira" is written in different ways, including differences in lower/upper cases and even wrong spelling.

For each column various inconsistencies were found, such as trailing white spaces. To fix these, a trimming transformation was applied to the columns:

$\Rightarrow$ **2.1** value.trim()

Other inconsistences, like words written in many alternative representations of the same object, were otherwise fixed using the clustering feature, which assembles words in groups based on heuristics methods.

Different methods were chosen based on the best obtained results. See Figure 2.7.

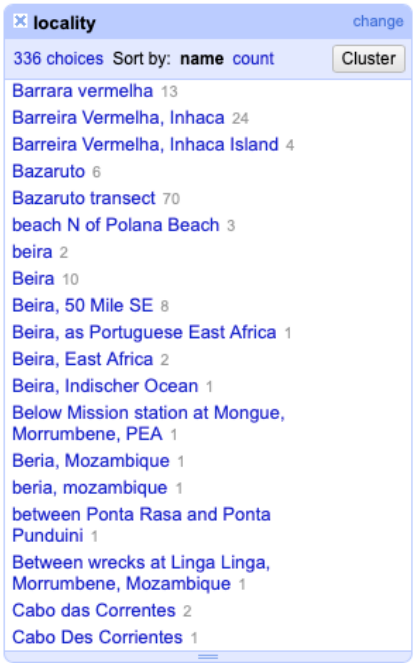

Figure 2.6 *– Part of the "Text facet" function for the locality column.*

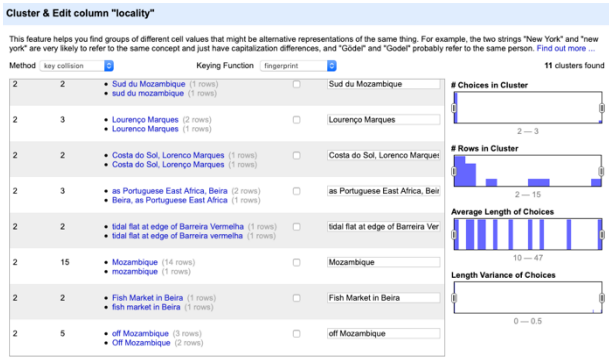

Figure 2.7 *– Part of the "Cluster" function for the locality column.*

b) Data Transformation

Specific columns were transformed, using the in-built OpenRefine language General Refine Expression Language (GREL).

Within the "depthInMeters" column, cells with more than one value (e.g. depth range) were split in two columns and their average was calculated as a new column value with "Column → Add column based on this column" menu option and then by using the GREL expression:

2.2 value.split("-")[1]

The entire values of the column "eventDate" were transformed in the "yyyy/mm/dd" short date format with the use of either of the following expressions:

2.3 toString(toDate(value),"yyyy/MM/dd")

or

2.4 value.slice(x, y) + '-' + value.slice(x, y) + '-' + value.slice(x, y)

For Uniform Resource Locator (URL) encoding type errors, the cells were corrected with the following expression:

2.5 value.unescape("html").unescape("xml")

Some latitude and longitude coordinates were in the Degrees, Minutes and Seconds (DMS) system and others in the decimal system. To convert all coordinates into the same decimal system, a Python script (coordDec.py) was coded for this particular purpose (See Section 2.1.5.), with the following mathematical formula:

2.6 Decimal Degrees = degrees + (minutes/60) + (seconds/3600)

c) Data Extension

Data in a dataset may be extended by using relational information, e.g. at the taxonomic level. The BioVeL extension was used to improve and enhance taxonomic classification information. The attributes kingdom, phylum, class, order and family were targeted against the checklist GBIF-Backbone (a publication that describes the taxonomy of a particular taxonomic group in a specific geographic region) to fill in missing data (GBIF Secretariat, 2017) See Figure 2.8 & 2.9.

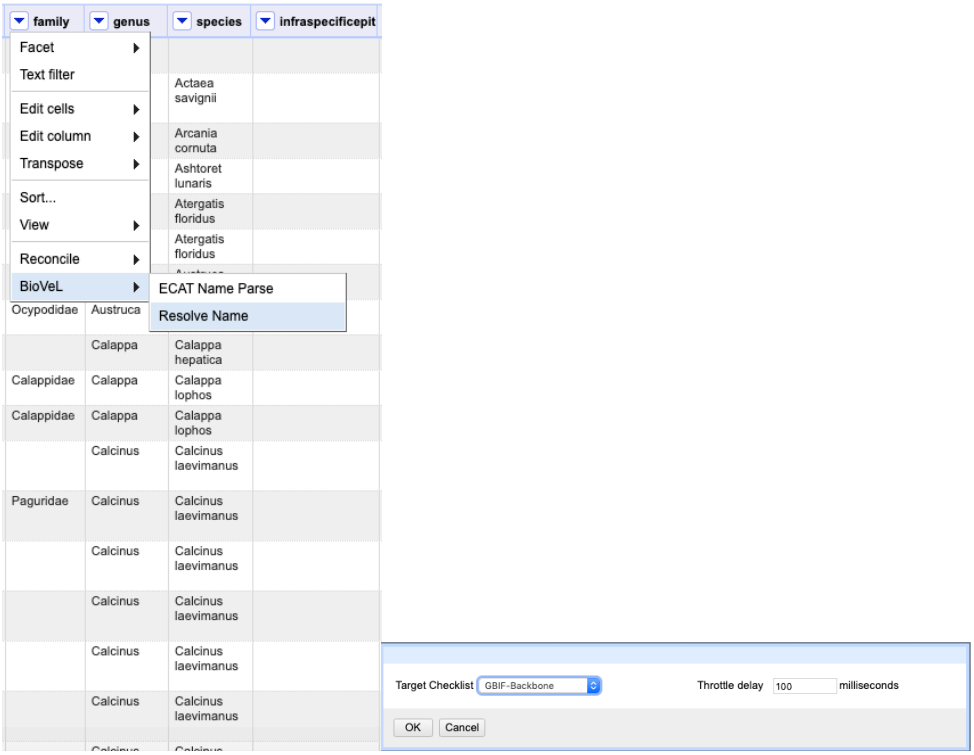

***Figure 2.8 & 2.9** – BioVeL path command and settings.*

2.1.5. Coding

A more efficient application of the digital tools used here, required the implementation of specific automating processes based on coding of distinct scripts, e.g. coding different scripts within the scope of each digital tool previously described, when necessary.

A number of scripts written in Python language were coded to assist several tasks of this study namely a) getData.py, b) previousName.py, c) taxonInfo.py, d) coordDec.py and e) matchShape.py. Python version 3.0 was used. See Section 6.2. for code scripts.

a) getData.py

The digital records of NHC from MNHN at https://science.mnhn.fr/institution/mnhn/collection/iu/item/list?order are not entirely available for download, they are displayed in HTML tables. This script parses the data in the HTML page into the correct rows and columns, avoiding a time-consuming task of manual text editing. An input allows the user to select the Order and Country to search through the API of the website:

```
13  Order = input("Insert ORDER: ")
14  Country = input("Insert COUNTRY: ")
15
16  search = "https://science.mnhn.fr/institution/mnhn/collection/iu/item/list?order="+Order.title()+"&country="+Country.upper()
```

The outcome of the request sent to the website is stored in a variable:

```
22    r = requests.get(search)
23    t = r.text
24
25    myfile = t.split("\n")
```

2

A loop is executed through the data stored in the variable and the corresponding variables are assigned the desired data:

```
18    links = []
19    dataSet = []
20    temp = "Name","Locality","Date","Depth","Coordinates"
```

Output is a CSV file with the mentioned variables as columns:

```
94    dataSet.append(temp)
95
96    with open('DataSet.csv', 'w') as csvFile:
97        writer = csv.writer(csvFile)
98        writer.writerows(dataSet)
99
100   csvFile.close()
```

b) previousName.py

Not all the data collected had the current accepted taxonomic name. This script confirms if the scientific name assigned to a certain species is still valid, otherwise it must be updated using the WoRMS website API. The scientific name column is used as input:

```
18  cl = Client('http://www.marinespecies.org/aphia.php?p=soap&wsdl=1')        # WoRMS webservice to feed with standard WoRMS data
31  with open(pathFile) as csv_file:
32      csv_reader = csv.reader(csv_file, delimiter=';')
33
34      for row in csv_reader:                                                  # loop to seek the input file
35
36          line_count+=1
37
38          if line_count > Prev_Run:                                           # calling the matchAphiaRecordsByNames function from Python
39
40              print('linha: %s' %(line_count-1))
41
42              st = 1
43
44              scinames = cl.factory.create('scientificnames')
45              scinames["_arrayType"] = "string[]"
46              scinames["scientificname"] = row[1]
47
48              try:
49                  array_of_results_array = cl.service.matchAphiaRecordsByNames(scinames, like="true", fuzzy="false", marine_only="true")
```

The code utilizes a loop to run through all of the scientificName entries and uses the API to verify if the name is still accepted. If not, the name presently accepted is written on the output:

```
57              for results_array in array_of_results_array:                    # loop to seek arrays in array for each entry
58                  for aphia_object in results_array:
59                      if aphia_object.status != "accepted":
60                              Prev_Name = (aphia_object.valid_name + " " + aphia_object.valid_authority)
61                      else:
62                          Prev_Name = ''
63                      if st == 1:
64                          Output.append((row[1], Prev_Name))                  # append the information retrieved into Output
65                          st=0
66
67  csv_file.close()
68
69  with open('previousName.csv', 'a+', encoding='utf-8') as csvFile:           # write csv file
70      writer = csv.writer(csvFile, delimiter=';')
71      writer.writerows(Output)
72
73  csvFile.close()
```

c) taxonInfo.py

This script utilizes the website API to retrieve the URL from the scientificName column to enrich the information in the dataset with information acquired from the website WoRMS:

```
18  cl = Client('http://www.marinespecies.org/aphia.php?p=soap&wsdl=1')         # WoRMS webservice to feed with standard WoRMS data
19
46              scinames = cl.factory.create('scientificnames')
47              scinames["_arrayType"] = "string[]"
48              scinames["scientificname"] = row[1]
49
50              try:
51                  array_of_results_array = cl.service.matchAphiaRecordsByNames(scinames, like="true", fuzzy="false", marine_only="false")
```

The code uses the dataset CSV file as input and runs a loop through all occurrences, using the API to make the correspondence between the scientificName and the species ID of the website:

```
33  with open(pathFile) as csv_file:
34      csv_reader = csv.reader(csv_file, delimiter=';')
35
36      for row in csv_reader:                                                  # loop to seek the inp
37
38          line_count+=1
39
40          if line_count > Prev_Run:                                           # calling the matchApl
41
```

As output, the CSV file has exactly the same columns as the input file with an added column with the URL of the website:

```
29              for results_array in array_of_results_array:                    # loop to seek arrays in array for each entry
30                  for aphia_object in results_array:
31                      if st == 1:
32                          Output.append((row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[11], row[12], row[13], row[14],
33                                         row[15], row[16], row[17], row[18], row[19], row[20], row[21], row[22], row[23], row[24], "http://www.marinespecies.org/aphia.php?p=taxdetails&id="+str(aphia_object.Aph
34                          st=0                                                # append the information retrieved into Output
35
36  csv_file.close()
37
38  with open('Result.csv', 'a+', encoding='utf-8') as csvFile:                 # write csv file
39      writer = csv.writer(csvFile, delimiter=';')
40      writer.writerows(Output)
41
42  csvFile.close()
```

d) coordDec.py

This script grants the multiple transformation of geographic coordinates from the DMS system to the decimal system. A text file containing the coordinates in the DMS format was used as an input file. A loop passes through these coordinates and the particular formula converts them into decimal coordinates given as output to a text file:

```
in_file = open(sys.argv[1], 'r')

out_file=open("Decimal.txt", 'w')

for line in in_file:

    c = line.split()

    D = float(c[0][:-1])

    M = 0
    if len(c) > 2:
        M = float(c[1][:-1])/60

    S = 0
    if len(c) > 3:
        S = float(c[2][:-2])/3600

    Dec = D + M + S

    if c[len(c)-1] == 'S' or c[len(c)-1] == 'W':
        Dec = 0 - Dec

    out_file.write("%.4f" % Dec+"\n")
```

e) matchShape.py

This script characterizes the habitat of each occurrence through proximity association between different shapefile layers. The shapefiles of each habitat are used as inputs, as are the occurrences of each country and the outlines of the countries:

```
# one input for each required file
pathFileIL = input("Insert shapefile Inland File Name or File Directory: ")

pathFileMG = input("Insert shapefile Mangrove File Name or File Directory: ")

pathFileMGpt = input("Insert shapefile Mangrove Points File Name or File Directory: ")

pathFileSG = input("Insert shapefile Seagrass Layer File Name or File Directory: ")

pathFileSGpt = input("Insert shapefile Seagrass Points File Name or File Directory: ")

pathFileCR = input("Insert shapefile Coral File Name or File Directory: ")

pathFileCRpt = input("Insert shapefile Coral Points File Name or File Directory: ")

pathFileDS = input("Insert Dataset File Name or File Directory: ")


shp = shapefile.Reader(pathFileIL) #open the shapefile
Inland = shp.shapes() # get all the polygons

shp = shapefile.Reader(pathFileMG) #open the shapefile
MG = shp.shapes() # get all the polygons

shp = shapefile.Reader(pathFileMG) #open the shapefile
MGpt = shp.shapes() # get all the points

shp = shapefile.Reader(pathFileSG) #open the shapefile
SG = shp.shapes() # get all the polygons

shp = shapefile.Reader(pathFileSGpt) #open the shapefile
SGpt = shp.shapes() # get all the points

shp = shapefile.Reader(pathFileCR) #open the shapefile
CR = shp.shapes() # get all the polygons

shp = shapefile.Reader(pathFileCR) #open the shapefile
CRpt = shp.shapes() # get all the points
```

One by one, the occurrences are compared with the habitat data layers/points and assigned to the closest habitat within a certain distance (chosen by the user). If not matched with any habitat layer, the result is either Coastal Area or Open Sea:

```
for row in csv_reader:
    if line_count == 0:
        line_count += 1
    else:
        pt = Point(float(row[17]), float(row[16]))
        print("Ponto #", (row[0], row[17], row[16]))
        In = 0

        if In == 0:
            print("Layer: Coral")
            for i in (CR):
                if (shape(i).distance(pt)) < 0.005: # checks whether the coordinates points from the dataset are at chosen distance (=0.005) from coral polygons
                    Output.append((row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[11], row[12], row[13], "Coral",
                        row[15], row[16], row[17], row[18], row[19], row[20], row[21], row[22], row[23], row[24]))
                    In = 1
                    break

        if In ==0:
            print("Layer: Costal Area")
            for i in (Inland):
                if (pt).within(shape(i)): # checks whether the coordinates points from the dataset are at chosen distance (=0.005) from seagrass points and appends the
                    Output.append((row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[11], row[12], row[13], "Costal Area",
                        row[15], row[16], row[17], row[18], row[19], row[20], row[21], row[22], row[23], row[24]))
                    In = 1
                    break

        if In == 0:
            print("Open Sea") # appends the result to the habitat column
            Output.append((row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[11], row[12], row[13], "Open Sea",
                row[15], row[16], row[17], row[18], row[19], row[20], row[21], row[22], row[23], row[24]))
```

The output is a CSV file with the habitat column filled with the assigned habitats (Coral, Mangrove, Seagrass, Coastal Area and Open Sea):

```
with open('MZ-Habitats.csv', 'w') as csvFile:           # write results in CSV file
    writer = csv.writer(csvFile, delimiter=';')
    writer.writerows(Output)

csvFile.close()
```

2.2. Data Management

Management and processing of biodiversity data from biological research collections, is much more effective through the use of a RDBMS. For this purpose, a relational database was implemented for the compiled data using the open source RDBMS PostgreSQL.

2.2.1. PostgreSQL

The complete dataset mentioned in Section 2.1. was imported into a PostgreSQL formatted database, by using the open-source administration and development platform for PostgreSQL, pgAdmin. PostgreSQL ran in version 11 and pgAdmin in version 4.9.

The dataset data was imported into four relational tables (occurrences, species, location and source) as shown below:

```
Create TABLE occurrences(
catalogNumber integer,
depthinMeters float,
individualCount integer not null,
eventDate date,,
PRIMARY KEY (catalogNumber)
);

Create TABLE species(
catalogNumber integer,
scientificName text not null,
kingdom text not null,
phylum text not null,
class text not null,
“order” text not null,
family text not null,
genus text not null,
species text,
infraspecificepithet text,
taxonRank text not  null,
previousName text,
taxonInfo text not null,
“type” text not null,
PRIMARY KEY (catalogNumber),
CONSTRAINT fk_sp FOREIGN KEY (catalogNumber) REFERENCES occurrences(catalogNumber)
);
```

```
Create TABLE location(
catalogNumber integer,
country text not null,
locality text not null,
verbatimLocality text not null,
habitat text not null,
decimalLatitude float not null,
decimalLongitude float not null,
geodeticDatum text not null,
PRIMARY KEY (catalogNumber),
CONSTRAINT fk_lo FOREIGN KEY (catalogNumber) REFERENCES occurrences(catalogNumber)
);

Create TABLE source(
catalogNumber integer,
institutionCode text,
occurrenceID text,
“references” text,
PRIMARY KEY (catalogNumber),
CONSTRAINT fk_so FOREIGN KEY (catalogNumber) REFERENCES occurrences(catalogNumber)
);
```

One additional column was created, named “type” in the table species. The “type” column, to be used in the symbology of the QGIS map, classifies the species occurrences in either crab, lobster or shrimp (See Section 2.3.1.f).

The columns “order”, “references” and “type” were created with quotation marks since there are in-built expressions of PostgreSQL with the same names. They were edited out after the table was created.

To set up a geodatabase, the PostGIS spatial extension was loaded:

```
CREATE EXTENSION postgis
```

Latitude and longitude coordinates were transformed into geospatial data with the use of SQL queries – to add a new column named coordinates to the table and transform the decimal latitude and longitude coordinates into the new column as geometry type, as presented:

```
Alter table location add column coordinates geometry (Point, 4326);

UPDATE location SET coordinates = ST_SetSRID(ST_MakePoint(
decimallongitude, decimallatitude), 4326);
```

In order to be used in the development of the web map (See Section 4.4.1.), a view of the column catalognumber and the concatenation of two columns, scientificname and catalognumber, was created, as shown below:

```
CREATE VIEW search AS
SELECT COALESCE(scientificname) || COALESCE('(#') || COALESCE(catalognumber) || COALESCE(')')
as scientificname, species.catalognumber from species;
```

The import into the database of all vector layers used in mapping was accomplished through the QGIS DB Manager.

2.3. Geographic Data Representation

2.3.1. QGIS

QGIS, is an open-source GIS, thus it was used for viewing, editing and analyzing the collected geographic data and information. QGIS version 3.4.9 was used since it was the most recent stable, long term release. Geographic analysis, using QGIS, entailed several steps such as a) geographical data processing, b) set up habitat layers, c) merging different habitat layers, d) adding background layers and e) adapting existent symbology.

a) Data Processing

The Excel file, with latitude and longitude coordinates stored in separated columns, was converted to CSV and then uploaded to QGIS as spatial layer through "Delimited Text" function. Point coordinates' longitude and latitude were mapped to X and Y fields, respectively. The coordinate reference system used was EPSG:4326 (or WGS 84). See Figure 2.10.

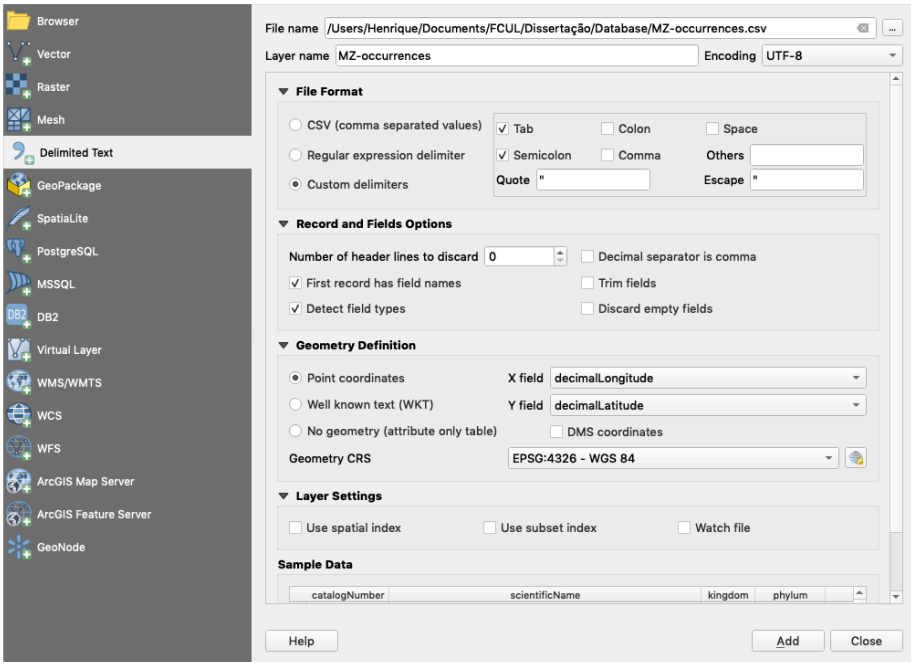

***Figure 2.10** – Depiction of an input used in the "Add Delimited Text Layer" sub-menu.*

The spatial layer with occurrences data-points was visualized with other two spatial layers for both studied countries, MOZ and STP, and their respective Exclusive Economic Zone (EEZ) (See Section 2.3.1.d). The data-points not assigned to either countries coastal zone or their EEZ, were removed from the dataset. The data-points were imported, and added as vector layer, through a direct connection between GIS and the PostgreSQL server.

b) Habitat Layers

The habitats studied encompassed corals, mangroves and seagrasses present in the coastal zones of MOZ and STP. The spatial datasets mapping the costal habitats, added as layers, were downloaded from the UN Environment

World Conservation Monitoring Centre website at http://data.unep-wcmc.org and the ReefBase website at http://reefbase.org/gis_maps/datasets.aspx. Sources used are listed in Section 6.1

The datasets used for each of the habitats were as follows:

- Coral
  - **Global Distribution of Coral Reefs (Dataset ID: WCMC-008):** the dataset shows the global distribution of coral reefs in tropical and subtropical regions, composed of one set of polygon occurrence data, with temporal range from 1954 to 2018 and reference system WGS 1984. Data retrieved on 23 March 2019 (version 4.0 - November 2018).
  - **Coral Bleaching (Dataset: ReefBase):** the dataset provides point occurrence data of observation details of coral bleaching around the world, with temporal range since early 2002. Data retrieved on 26 May 2019.
  - **Monitoring Sites (Dataset: ReefBase):** the dataset provides point occurrence data on coral reef monitoring sites locations from major reef monitoring programs. Data retrieved on 26 May 2019.
  - **Reefs Location (Dataset: ReefBase):** the dataset provides point occurrence data on coral reef locations. Data retrieved on 26 May 2019.
  - **Marine Protected Areas (Dataset: ReefBase):** the dataset provides point occurrence data on marine protected locations with coral reefs areas. Data retrieved on 26 May 2019.
- Mangrove
  - **World Atlas of Mangroves (Dataset ID: WCMC-011):** the dataset shows the global distribution of mangroves, and was produced mostly from satellite imagery, composed of one set of polygon occurrence data, with temporal range mainly from 1999 to 2003 and reference system WGS 1984 (version 2.0 – December 2017).
  - **Global Distribution of Mangroves USGS (Dataset ID: WCMC 010):** the dataset shows the global distribution of mangrove forests, derived from earth observation satellite imagery, composed of one set of polygon occurrence data, with temporal range from 1997 to 2000 and reference system WGS 1984. Data retrieved on 4 April 2019 (version 1.3 – June 2015).
  - **Global Mangrove Watch (Dataset ID: GMW-001):** the dataset shows a global baseline map of mangroves using satellite imagery, composed of one set of polygon occurrence data, with temporal range from 1996 to 2016. Data retrieved on 4 April 2019 (version 2.0).
- Seagrass
  - **Global Distribution of Seagrasses (Dataset ID: WCMC-013-014):** the dataset shows the global distribution of

seagrasses, composed of two subsets of point and polygon occurrence data, with temporal range from 1934 to 2015 and reference system WGS 1984. Data retrieved on 23 March 2019 (version 6.0 - June 2018).

To fill the lack of information on habitats in São Tomé and Príncipe, the search was extended to scientific literature. Sources used are listed in Section 6.1.

The collected text data was added to QGIS as delimited text layers and saved, for each habitat, as ESRI Shapefiles with CRS EPSG:4326 – WGS 84.

Data collected from images depicting habitats, was added to QGIS as Raster Layers and then georeferenced. Georeferencing is the process of taking a raster image or vector coverage, assigning it a coordinate system and coordinates, and translating, transforming, and warping it into position relative to some other spatial data. For that the inbuilt QGIS Georeferencer function was used. See Figure 2.11 and 2.12.

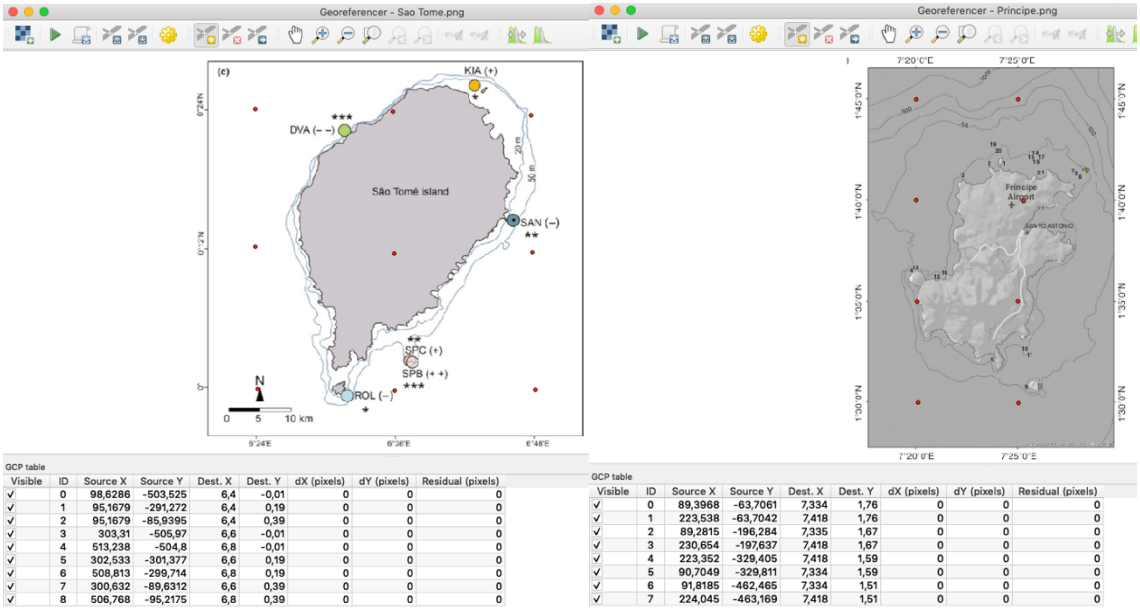

Figure 2.11 & 2.12 – *QGIS Georeferencer interface using two raster data collected from literature with Mangrove habitats (identified by the colored circles in the left image and the numbers in the right image). Raster Ground Control Points are marked as red dots.*

The process of georeferencing was accomplished by assigning real-world coordinates to specific pixels of the raster, obtained by the coordinates from the markings on the map image itself. For georeferencing a total of 9 ground control points were used in the raster relative to São Tomé Island and 8 ground control points for the raster relative to Príncipe Island. Through the use of these sample coordinates the georeferenced rasters were warped, added to QGIS as delimited text layers, saved as ESRI Shapefiles and made to fit within the chosen coordinate system CRS EPSG:4326 – WGS 84.

c) Merging Layers

In order to combine all data corresponding to each habitat (Corals, Mangroves and Seagrasses) in one shapefile only, the layers with the same geometry type e.g. “Point” or “Polygon” were merged into a single layer using the following path command:

Vector → Data Management Tools → Merge Vector Layers

The input layers “Global Distribution of Coral Reefs”, “Coral Bleaching”, “Monitoring Sites”, “Reefs Location” and “Marine Protected Areas” were merged into a point data layer named “Coral point-data”. Both input layers “Global Distribution of Mangroves USGS” and “Global Mangrove Watch” were merged into a polygon data layer named “Mangrove polygon-data”. All layers created manually were also joined to their respective habitat layers.

The attribute table of the resulting layer contained the attributes from all input layers, adding all the different attributes.

For layers with global information, only features of interest were selected, corresponding to the studied countries and/or surrounding African coastal areas.

d) Background Layers

All background layers were downloaded from public domain map data available online:

- From Natural Earth website at https://www.naturalearthdata.com the vector layer “Coastline” with large scale data, 1:10m.
- From Map Library website at http://www.maplibrary.org/library/index.htm the administrative boundaries, divisions and outline of MOZ and STP as ESRI Shapefile format.
- From Marine Regions website at http://www.marineregions.org the Mozambican and São Tomé and Principe EEZ as ESRI Shapefile format.

e) PostGIS Layers

To add from the database the previously created tables (occurrences, species, location and source), view (search) and imported vector layers, the following path command was used:

Layer → Add Layer → Add PostGIS Layers

The tables and view with no geometry (occurrences, species, source and search) were joined into the table with spatial type point (location). See Figure 2.13.

Figure 2.13 *– Attribute tables: occurrences, species and source joined to the vector layer: location.*

f) Symbology

The symbology of the layers was changed to be more informative in helping distinguish different information. Relative to the species occurrences layer, a Rule-based style was utilized, using the columns individualCount and type to classify, respectively, two rules: 1) a graduated style by size and 2) a different symbol by species-likeness. Cases in which the individualCount was null, were assumed to be 1. The symbols were also changed with scalable vector graphics (SVG) to be more visually appealing. See Figure 2.14.

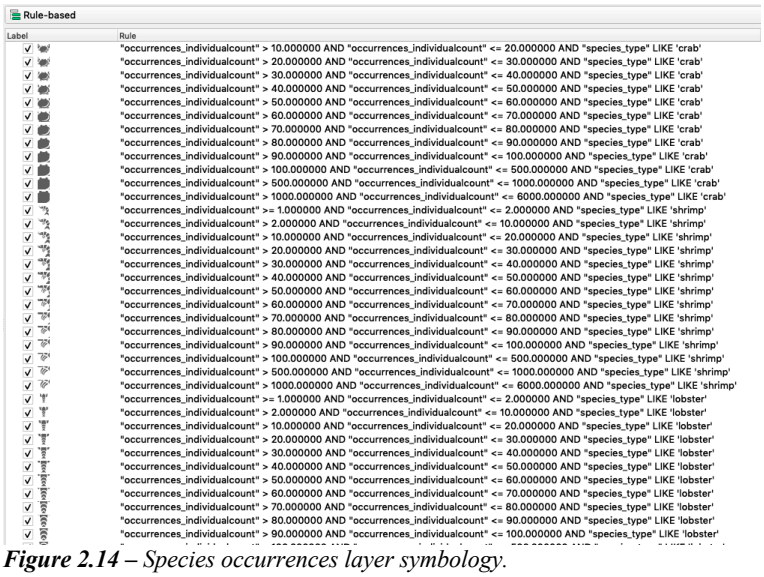

Figure 2.14 *– Species occurrences layer symbology.*

2.4. Online Data Dissemination

To promote online data dissemination and make it user-friendly, a digital platform map was developed using the Leaflet open-source JavaScript library designed specifically for web mapping.

2.4.1. Leaflet

In order to build a web map based on the QGIS project, the qgis2web plugin for QGIS was used to generate a Leaflet interactive web map (Gandhi, 2019).

Foremost, the QGIS project was reformed to produce a user-friendly visualization. Under the Attributes Form, not all attribute fields were kept for display, some were hidden, so that the user is not able to see its contents (e.g. verbatimLocality). Alternatives names were also defined for the fields, under the Alias column, instead of the DwC terms, for a better user comprehension. The background color of the canvas was changed as well for a more natural looking color in the marine distribution map.

To achieve a better performance of the web map, heavier shapefiles were edited with the software Mapshaper at www.mapshaper.org, in order to simplify the layers.

In the making of the Leaflet map itself, the following settings were used, see Figure 2.15 & 2.16:

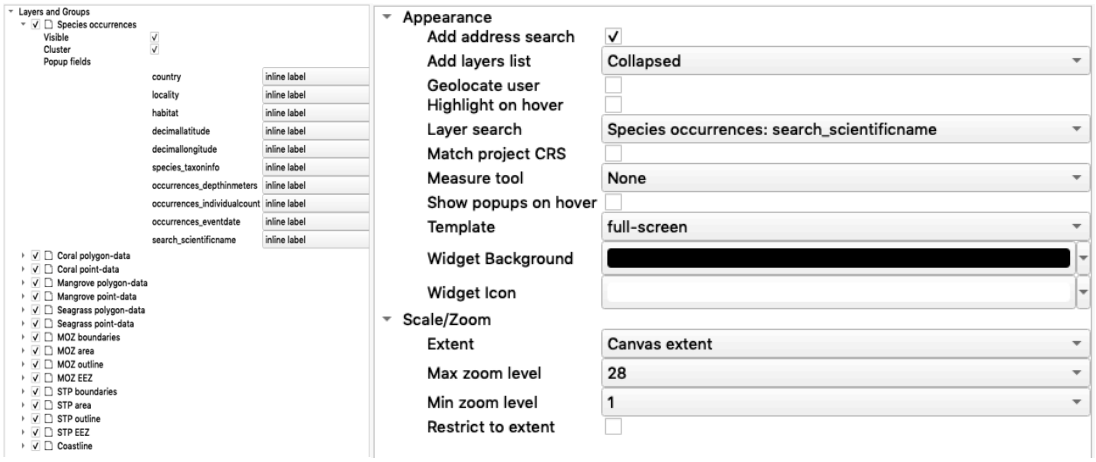

Figure 2.15 & 2.16 – *Leaftlet web map settings on layers, groups, appearance, scale and zoom.*

a) Customization

In addition to the tools implemented in the basic application, others were also added to the Leaflet script. See Section 6.2. for code script

- Opacity was added to all the different symbology in the species occurrences layer:

```
    // Start of if blocks and style check logic
    if (Speciesoccurrences_15rule0_eval_expression(context)) {
        return {
        pane: 'pane_Speciesoccurrences_15',
        opacity: 0.7,
rotationAngle: 0.0,
rotationOrigin: 'center center',
icon: L.icon({
    iconUrl: 'markers/crab-2.svg',
    iconSize: [19.0, 19.0]
```

- The layers list has been altered to preserve a concise and simple symbology legend:

```
L.control.layers(baseMaps,{'Species occurrences<br/><table><tr><td style="text-align: center;"><img src="legend/Speciesoccurrences_15_0.png";"><img src="legend/
    Speciesoccurrences_15_28.png";"><img src="legend/Speciesoccurrences_15_14.png" /></td><td></td></tr><tr><td style="text-align:
    center;"></td><td></td></tr></table>': cluster_Speciesoccurrences_15,'<img src="legend/Coralpolygondata_14.png" /> Coral polygon-data':
    layer_Coralpolygondata_14,'<img src="legend/Coralpointdata_13.png" /> Coral point-data': cluster_Coralpointdata_13,'<img src="legend/
    Mangrovepolygondata_12.png" /> Mangrove polygon-data': layer_Mangrovepolygondata_12,'<img src="legend/Mangrovepointdata_11.png" /> Mangrove point-data':
    cluster_Mangrovepointdata_11,'<img src="legend/Seagrasspolygondata_10.png" /> Seagrass polygon-data': layer_Seagrasspolygondata_10,'<img src="legend/
    Seagrasspointdata_9.png" /> Seagrass point-data': cluster_Seagrasspointdata_9,'<img src="legend/MOZboundaries_8.png" /> MOZ boundaries': layer_MOZboundaries_8
    ,'<img src="legend/MOZarea_7.png" /> MOZ area': layer_MOZarea_7,'<img src="legend/MOZoutline_6.png" /> MOZ outline': layer_MOZoutline_6,'<img src="legend/
    MOZEEZ_5.png" /> MOZ EEZ': layer_MOZEEZ_5,'<img src="legend/STPboundaries_4.png" /> STP boundaries': layer_STPboundaries_4,'<img src="legend/STParea_3.png"
    /> STP area': layer_STParea_3,'<img src="legend/STPoutline_2.png" /> STP outline': layer_STPoutline_2,'<img src="legend/STPEEZ_1.png" /> STP EEZ':
    layer_STPEEZ_1,'<img src="legend/Coastline_0.png" /> Coastline': layer_Coastline_0,}).addTo(map);
```

2.5. Data Analysis

Data analysis was performed using the Power BI Desktop, to build data visualizations of the dataset for species richness, habitat distribution and sources participation.

3. Results

3.1. Biodiversity Dataset Construction

The structure of the dataset created during this study has 26 DwC attributes related e.g. to taxon level, locality, coordinates and habitat of occurrence, number of individuals and other relevant info. This structure matches the set of the pre-configured spreadsheet template prepared for publishing the dataset to GBIF.

The dataset comprises decapod crustacean occurrences data from mangroves, seagrasses, corals, open sea and other coastal (aggregated) habitats along marine areas of MOZ and STP, including all the occurrences data belonging to decapod crustacean, compiled from NHC records, digital repositories and scientific literature. In total, 7486 species occurrences entries were charted in both countries, comprising 102 decapod families and 793 species. See Figure 3.1.

catalogueNumbe	scientificName	kingdom	phylum	class	order	family	genus	species	infraspecificEpithe	taxonRank	previousName	country	locality	verbatimLocali	habitat	decimalLatitud	decimalLongitud	geodeticDatum
3652	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	n/a		Open Sea	-22.1167	35.7500	EPSG:4326
3653	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	Zambezi River	Zambeze transect	Open Sea	-19.5745	36.75733	EPSG:4326
3654	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	Zambezi River	Zambeze transect	Open Sea	-19.5745	36.75733	EPSG:4326
3655	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	Zambezi River	Zambeze transect	Open Sea	-19.5745	36.75733	EPSG:4326
3656	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	Zambezi River	Zambeze transect	Open Sea	-19.5745	36.75733	EPSG:4326
3657	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	Zambezi River	Zambeze transect	Open Sea	-19.5745	36.75733	EPSG:4326
3658	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	Zambezi River	Zambeze transect	Open Sea	-19.5885	36.762	EPSG:4326
3659	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	Zambezi River	Zambeze transect	Open Sea	-19.5885	36.762	EPSG:4326
3660	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	Zambezi River	Zambeze transect	Open Sea	-19.5885	36.762	EPSG:4326
3661	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	n/a		Open Sea	-19.45000076293	36.8499984741211	EPSG:4326
3662	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	n/a		Open Sea	-18.23329925537	37.5167007446289	EPSG:4326
3663	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	n/a		Open Sea	-17.54999923706	38.4500007629395	EPSG:4326
3664	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	n/a		Open Sea	-16.33329963684	40.13330078125	EPSG:4326
3665	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	n/a		Open Sea	-21.3667	35.6667	EPSG:4326
3666	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	n/a		Open Sea	-17.5500	38.4500	EPSG:4326
3667	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	n/a		Open Sea	-19.4500	36.8500	EPSG:4326
3668	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	n/a		Open Sea	-17.5500	38.4500	EPSG:4326
3669	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	n/a		Open Sea	-16.3333	40.1333	EPSG:4326
3670	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	n/a		Open Sea	-24.0333	35.7500	EPSG:4326
3671	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	n/a		Open Sea	-25.3333	35.2833	EPSG:4326
3672	Polycheles typhlo	Animalia	Arthropoda	Malacostrac	Decapoda	Polychelida	Polycheles	Polycheles typhlops		SPECIES		Mozambiqu	n/a		Open Sea	-25.3333	35.2833	EPSG:4326
3673	Polyonyx senegale	Animalia	Arthropoda	Malacostrac	Decapoda	Porcellanid	Polyonyx	Polyonyx senegalensis		SPECIES		Mozambiqu	n/a		Seagrass	-25.98329925537	32.7832984924316	EPSG:4326
3674	Polyonyx Stimpso	Animalia	Arthropoda	Malacostrac	Decapoda	Porcellanid	Polyonyx			GENUS		Mozambiqu	Zambezi River	Zambeze River, Off	Open Sea	-19.15	36.9167	EPSG:4326
3675	Pontophilus Leach	Animalia	Arthropoda	Malacostrac	Decapoda	Crangonida	Pontophilus			GENUS		Mozambiqu	n/a		Open Sea	-18.23329925537	37.5167007446289	EPSG:4326
3676	Portunus hastatus	Animalia	Arthropoda	Malacostrac	Decapoda	Potamidae	Portunus	Portunus hastatus		SPECIES		Sao Tome ar	Gulf of Guinea	Guinea-Bucht, St.	Open Sea	0.1	6.05	EPSG:4326
3677	Portunus pelagicu	Animalia	Arthropoda	Malacostrac	Decapoda	Potamidae	Portunus	Portunus pelagicus		SPECIES		Mozambiqu	Inhaca Island	Inhaca Island	Open Sea	-26.0104	32.9122	EPSG:4326
3678	Portunus pelagicu	Animalia	Arthropoda	Malacostrac	Decapoda	Potamidae	Portunus	Portunus pelagicus		SPECIES		Mozambiqu	Praia do Bilen	Praia Do Bilene: In	Coastal Area	-25.2863	33.2571	EPSG:4326
3679	Portunus pelagicu	Animalia	Arthropoda	Malacostrac	Decapoda	Potamidae	Portunus	Portunus pelagicus		SPECIES		Mozambiqu	Govuro	Govuro Mar Abert	Seagrass	-21.15137	35.08851	EPSG:4326
3680	Portunus pelagicu	Animalia	Arthropoda	Malacostrac	Decapoda	Potamidae	Portunus	Portunus pelagicus		SPECIES		Mozambiqu	Govuro	Govuro Estuário	Seagrass	-21.15137	35.08851	EPSG:4326
3681	Portunus pelagicu	Animalia	Arthropoda	Malacostrac	Decapoda	Potamidae	Portunus	Portunus pelagicus		SPECIES		Mozambiqu	Govuro	Govuro Mar Abert	Seagrass	-21.15137	35.08851	EPSG:4326
3682	Portunus pelagicu	Animalia	Arthropoda	Malacostrac	Decapoda	Potamidae	Portunus	Portunus pelagicus		SPECIES		Mozambiqu	Govuro	Govuro Mar Abert	Seagrass	-21.15137	35.08851	EPSG:4326
3683	Portunus pelagicu	Animalia	Arthropoda	Malacostrac	Decapoda	Potamidae	Portunus	Portunus pelagicus		SPECIES		Mozambiqu	Govuro	Govuro Estuário	Seagrass	-21.15137	35.08851	EPSG:4326
3684	Portunus pelagicu	Animalia	Arthropoda	Malacostrac	Decapoda	Potamidae	Portunus	Portunus pelagicus		SPECIES		Mozambiqu	Govuro	Govuro Mar Abert	Seagrass	-21.15137	35.08851	EPSG:4326

depthInMeters	individualCount	eventDate	institutionCod	occurrenceID	references	taxonInfo
500	1	09/06/1994	SAM	A41697	Crustacea Decapoda: Review of the genera and species of the family Polychelidae Wood-Mason, 1875	http://www.marinespecies.org/aphia.php?p=taxdetails&
444	1	13/04/2009	MNHN	http://coldb.mnhn.fr/catalognumber/mnhn/iu/2008-10178		http://www.marinespecies.org/aphia.php?p=taxdetails&
444	1	13/04/2009	MNHN	http://coldb.mnhn.fr/catalognumber/mnhn/iu/2008-10377		http://www.marinespecies.org/aphia.php?p=taxdetails&
444	1	13/04/2009	MNHN	http://coldb.mnhn.fr/catalognumber/mnhn/iu/2008-10379		http://www.marinespecies.org/aphia.php?p=taxdetails&
444	1	13/04/2009	MNHN	http://coldb.mnhn.fr/catalognumber/mnhn/iu/2008-10381		http://www.marinespecies.org/aphia.php?p=taxdetails&
444	1	13/04/2009	MNHN	http://coldb.mnhn.fr/catalognumber/mnhn/iu/2013-7377		http://www.marinespecies.org/aphia.php?p=taxdetails&
521	1	13/04/2009	MNHN	http://coldb.mnhn.fr/catalognumber/mnhn/iu/2008-10382		http://www.marinespecies.org/aphia.php?p=taxdetails&
521	1	13/04/2009	MNHN	http://coldb.mnhn.fr/catalognumber/mnhn/iu/2008-10383		http://www.marinespecies.org/aphia.php?p=taxdetails&
521	1	13/04/2009	MNHN	http://coldb.mnhn.fr/catalognumber/mnhn/iu/2013-7374		http://www.marinespecies.org/aphia.php?p=taxdetails&
486	1	14/06/1994	SAMC	SAMC-MBR-010780		http://www.marinespecies.org/aphia.php?p=taxdetails&
472	1	17/06/1994	SAMC	SAMC-MBR-010149		http://www.marinespecies.org/aphia.php?p=taxdetails&
433	1	15/06/1994	SAMC	SAMC-MBR-010177		http://www.marinespecies.org/aphia.php?p=taxdetails&
500	1	16/06/1994	SAMC	SAMC-MBR-010146		http://www.marinespecies.org/aphia.php?p=taxdetails&
481	4	13/06/1994	SAM	A41698	Crustacea Decapoda: Review of the genera and species of the family Polychelidae Wood-Mason, 1876	http://www.marinespecies.org/aphia.php?p=taxdetails&
433	1	14/06/1994			Crustacea Decapoda: Review of the genera and species of the family Polychelidae Wood-Mason, 1877	http://www.marinespecies.org/aphia.php?p=taxdetails&
486	2	14/06/1994	SAM	A41703	Crustacea Decapoda: Review of the genera and species of the family Polychelidae Wood-Mason, 1874	http://www.marinespecies.org/aphia.php?p=taxdetails&
433	1	15/06/1993	SAM	A41704	Crustacea Decapoda: Review of the genera and species of the family Polychelidae Wood-Mason, 1879	http://www.marinespecies.org/aphia.php?p=taxdetails&
500	1	16/06/1994	SAM	A41656	Crustacea Decapoda: Review of the genera and species of the family Polychelidae Wood-Mason, 1880	http://www.marinespecies.org/aphia.php?p=taxdetails&
517	2	20/06/1995	SAM	A41695	Crustacea Decapoda: Review of the genera and species of the family Polychelidae Wood-Mason, 1882	http://www.marinespecies.org/aphia.php?p=taxdetails&
557.5	9	21/02/1951	ZMC		Crustacea Decapoda: Review of the genera and species of the family Polychelidae Wood-Mason, 1884	http://www.marinespecies.org/aphia.php?p=taxdetails&
705	1	21/02/1951	ZMC		Crustacea Decapoda: Review of the genera and species of the family Polychelidae Wood-Mason, 1885	http://www.marinespecies.org/aphia.php?p=taxdetails&
	1		SAMC	SAMC-MBR-003444		http://www.marinespecies.org/aphia.php?p=taxdetails&
88	1	09/10/1964	USNM	http://n2t.net/ark:/65665/3e4cc587d-238a-4a88-9360-31fb42064243		http://www.marinespecies.org/aphia.php?p=taxdetails&
472	1	17/06/1994	SAMC	SAMC-MBR-009987		http://www.marinespecies.org/aphia.php?p=taxdetails&
24	1		ZMH			http://www.marinespecies.org/aphia.php?p=taxdetails&
	1	02/10/2006	iNaturalist	https://www.inaturalist.org/observations/11044277		http://www.marinespecies.org/aphia.php?p=taxdetails&
	1	04/04/2015	iNaturalist	https://www.inaturalist.org/observations/11066625		http://www.marinespecies.org/aphia.php?p=taxdetails&
	1	10/04/2009	IIP	urn:fish:IIP:MZ:Inhambane:3478		http://www.marinespecies.org/aphia.php?p=taxdetails&
	1	25/11/2011	IIP	urn:fish:IIP:MZ:Inhambane:41352		http://www.marinespecies.org/aphia.php?p=taxdetails&
	1	09/03/2009	IIP	urn:fish:IIP:MZ:Inhambane:2031		http://www.marinespecies.org/aphia.php?p=taxdetails&
	1	16/04/2009	IIP	urn:fish:IIP:MZ:Inhambane:3725		http://www.marinespecies.org/aphia.php?p=taxdetails&
	1	25/02/2016	IIP	urn:fish:IIP:MZ:Inhambane:86831		http://www.marinespecies.org/aphia.php?p=taxdetails&
	1	05/05/2009	IIP	urn:fish:IIP:MZ:Inhambane:4692		http://www.marinespecies.org/aphia.php?p=taxdetails&

Figure 3.1 *– Dataset section showing its structure with the attribute column terms and occurrences examples.*

These data were further analyzed to determine the number of occurrences and distribution per country through different habitats.

Regarding the sources by type, the most representative, by a large amount, were the digital repositories with 6308 (84.2%) species occurrences, NHC with 981 (13.1%) species occurrences and scientific literature with 199 (2.7%) species occurrences. See Figure 3.2.

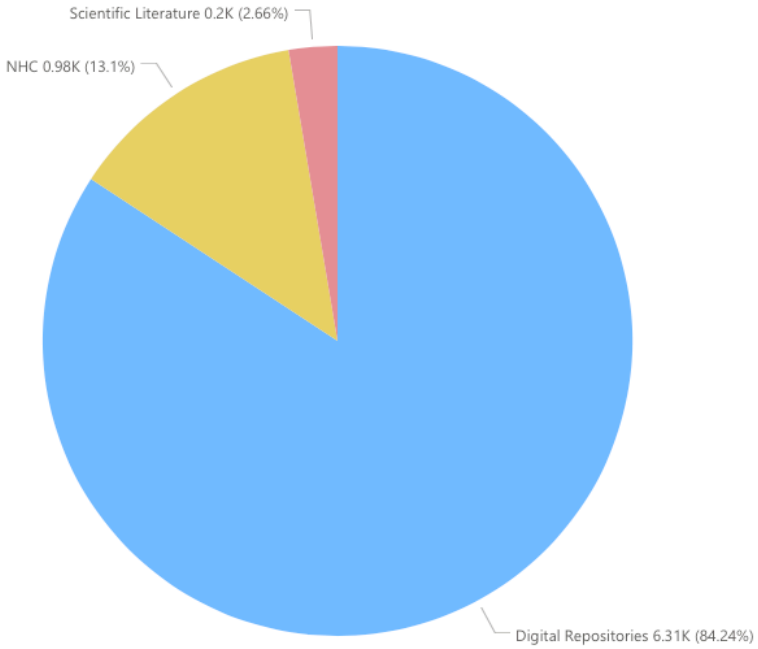

Figure 3.2 *– Representative sources by numbers and percentages.*

Regarding the institutions sources, the most representative, by a large amount, was the National Institute of Fisheries Research (IIP) with 5320 (72.4%) species occurrences, followed by MNHN with 741 (10.1%) species occurrences. A large number of institutions was represented by only one occurrence, such as the Zoological Museum of the Moscow State University (ZMMU) and the South African Museum. See Figure 3.3.

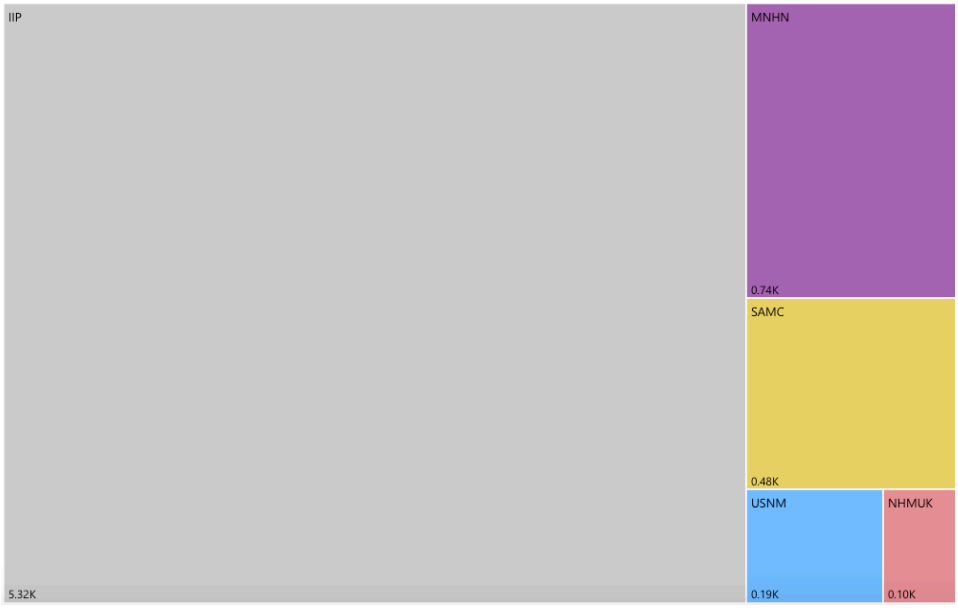

Figure 3.3 *– Top 5 most represented institutions sources in the dataset.*

Of the 102 decapod families recorded in this study, only 33 families (32.3%) were represented in STP, assembled into 70 species. The majority of the recorded decapod families

(97%) were present in MOZ, assembled into 723 species. The other 3% (families *Coenobitidae, Munidopsidae* and *Plagusiidae*) weren't found in MOZ but were recorded for STP.

Considering species occurrences across the 5 habitats defined in this study: 210 (2.8%) were found in coastal areas, 20 (2.7%) in coral, 131 (1.75%) in mangroves, 1516 (20.3%) in open sea and 5429 (72.5%) in seagrasses. See Figure 3.4.

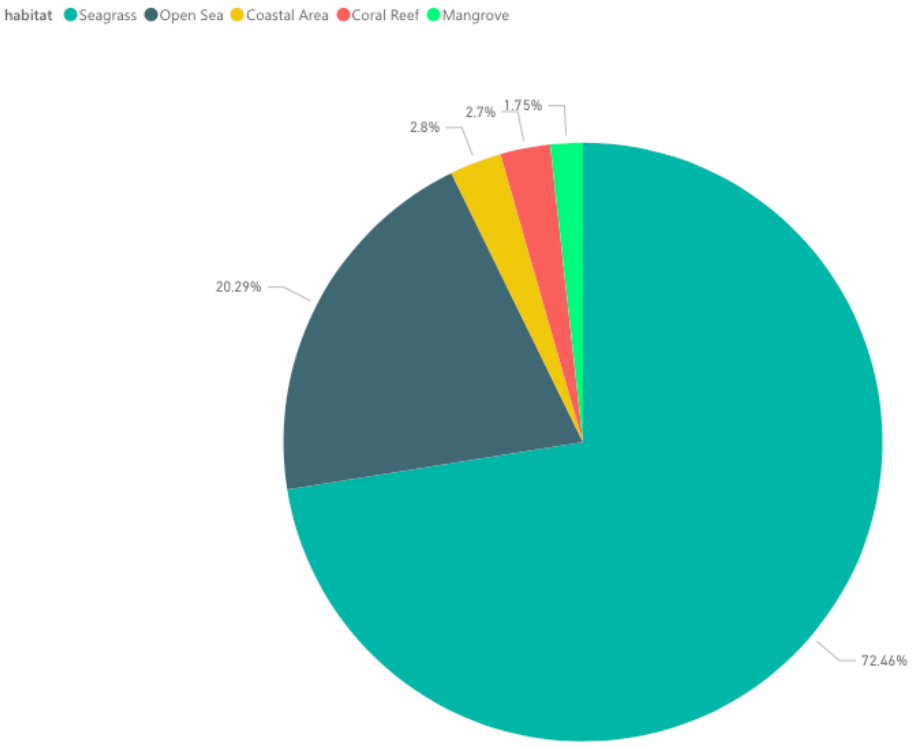

***Figure 3.4** – Overall percentage of species occurrences by habitat type.*

Considering the 5 habitats per country, in MOZ there were 197 (2.6%) species occurrences in coastal areas, 194 (2.6%) in corals, 91 (1.2%) in mangroves, 1471 (20%) in open sea and 5392 (73.4%) in seagrasses. In STP there were 13 (8.67%) species occurrences in coastal areas, 8 (5.3%) in corals, 40 (26.6%) in mangroves, 50 (33.3%) in open sea and 39 (26.7%) in seagrasses. See Figure 3.5.

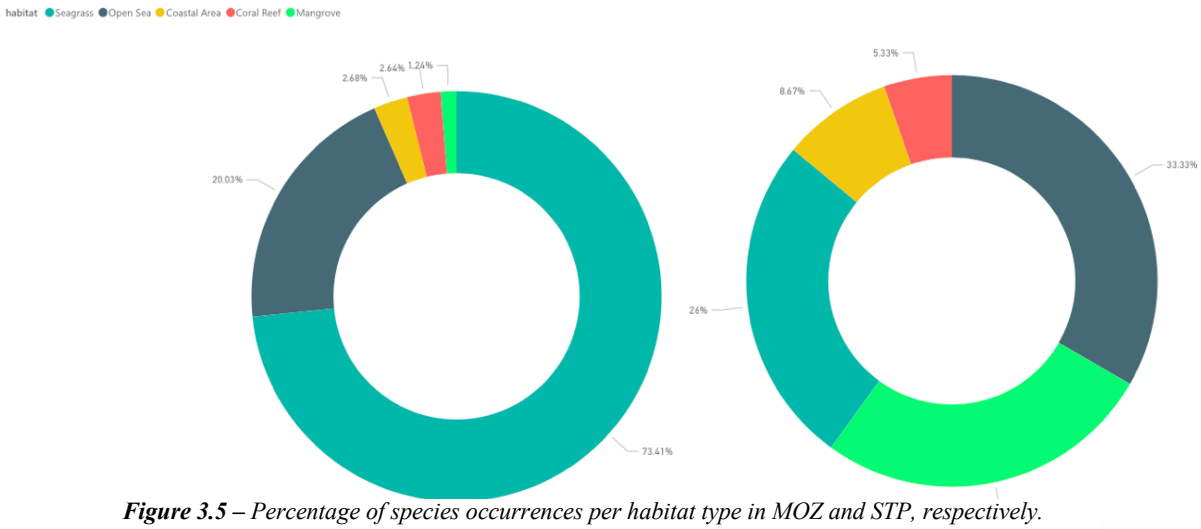

***Figure 3.5** – Percentage of species occurrences per habitat type in MOZ and STP, respectively.*

Of all the decapods families found in the coastal zones of MOZ and STP, the Potamidae family with 3734 (49.9%) species occurrences was the most well represented, followed by the Penaeidae family with 1582 (21.1%) species occurrences. A large number of families was represented by only one species occurrence such as e.g. Thiidae and Sotoplacidae. In MOZ the former two families, Potamidae and Penaeidae, also presented most species occurrences. See Figure 3.6.

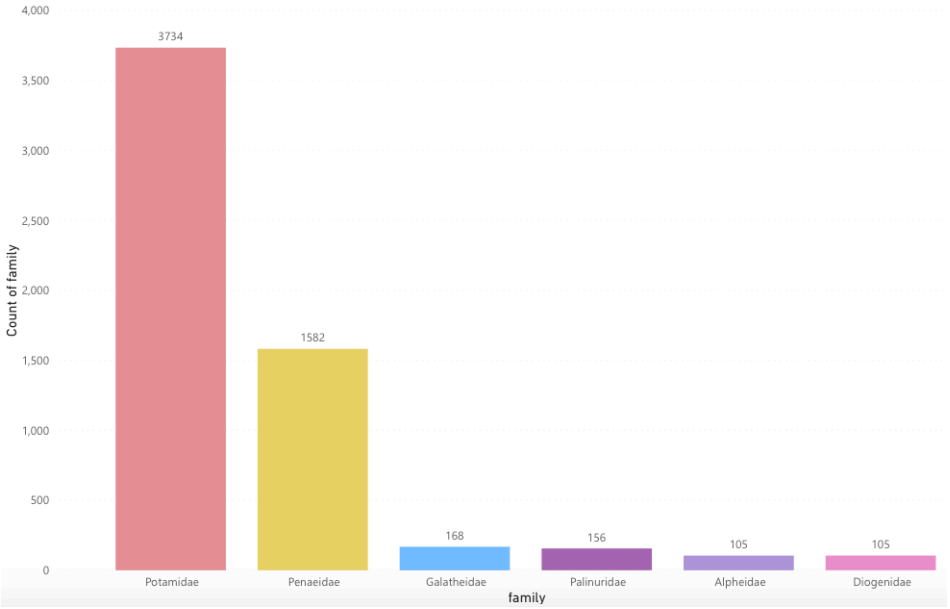

***Figure 3.6** – Top 6 most represented families in both countries with greater than or equal to 100 occurrences.*

In STP, the Alpheidae family was the most frequent with 46 (30.7%) species occurrences, followed by Sergestidae with 18 (12%) species occurrences. The least frequent families, for example Upogebiidae and Trapeziidae, were represented by only one species occurrence. See Figure 3.7.

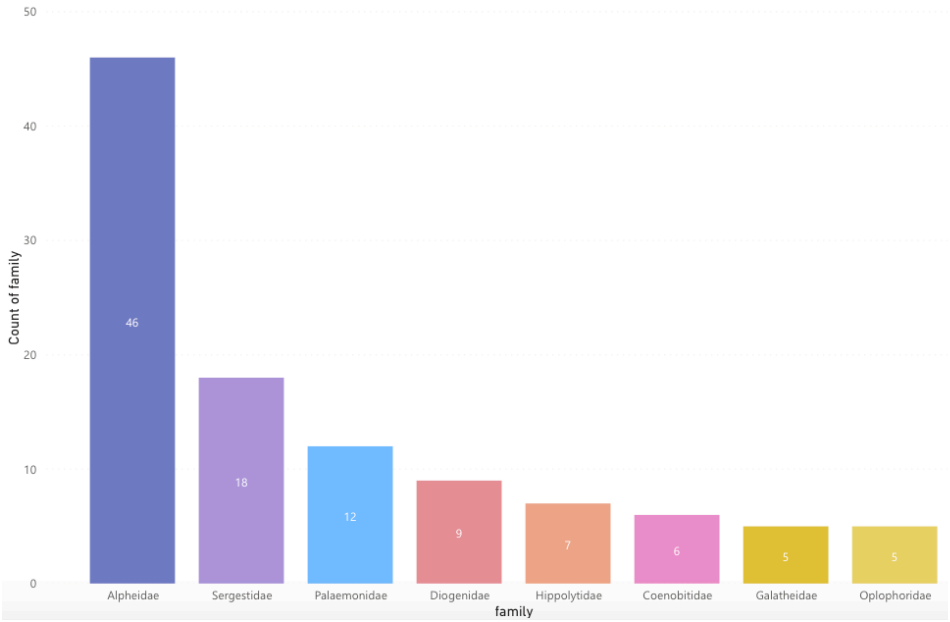

***Figure 3.7** – Top 8 most represented families in STP with greater than or equal to 5 occurrences.*

Regarding the occurrence per species, *Portunus pelagicus* with 2589 (36.1%) occurrences was the most representative species, followed by *Penaeus indicus* with 609 (8.5%) occurrences. A large number of species was represented by only one occurrence such as *Zenopontonia rex* and *Xenophthalmodes moebii*. In MOZ these two species, *P. pelagicus* and *P. indicus*, had most species occurrences as well. See Figure 3.8.

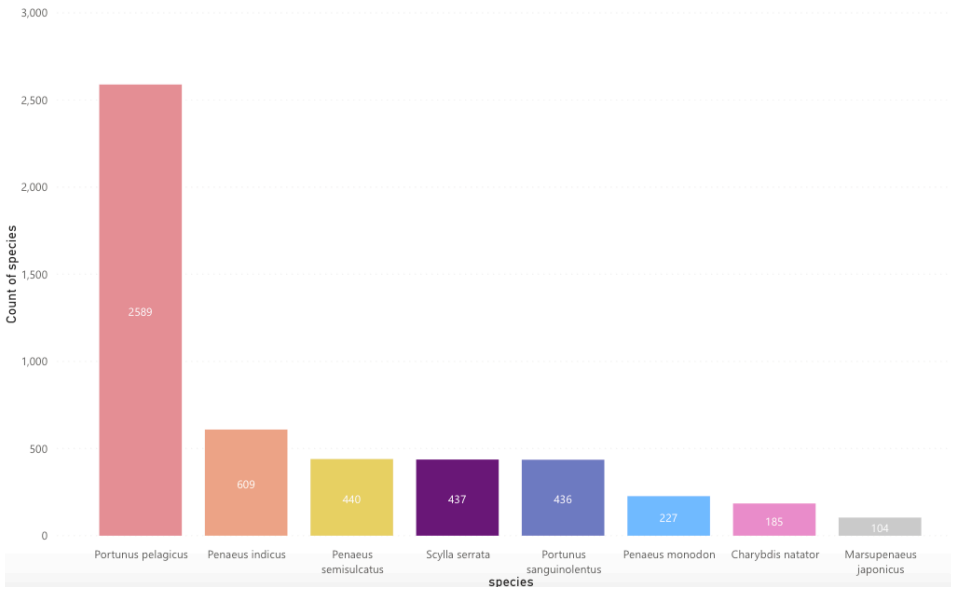

***Figure 3.8** – Top 8 most represented species occurrences in Mozambique and São Tomé and Príncipe with greater than or equal to 100 occurrences.*

In STP, the most representative species was *Alpheus fagei* with 9 (6.5%) occurrences, followed by both *Alpheus vanderbilti and Synalpheus africanus* with 7 (5%) occurrences. A large number of species was represented by only one occurrence, among them *Troglocarcinus corallicola* and *Systellaspis debilis*. See Figure 3.9.

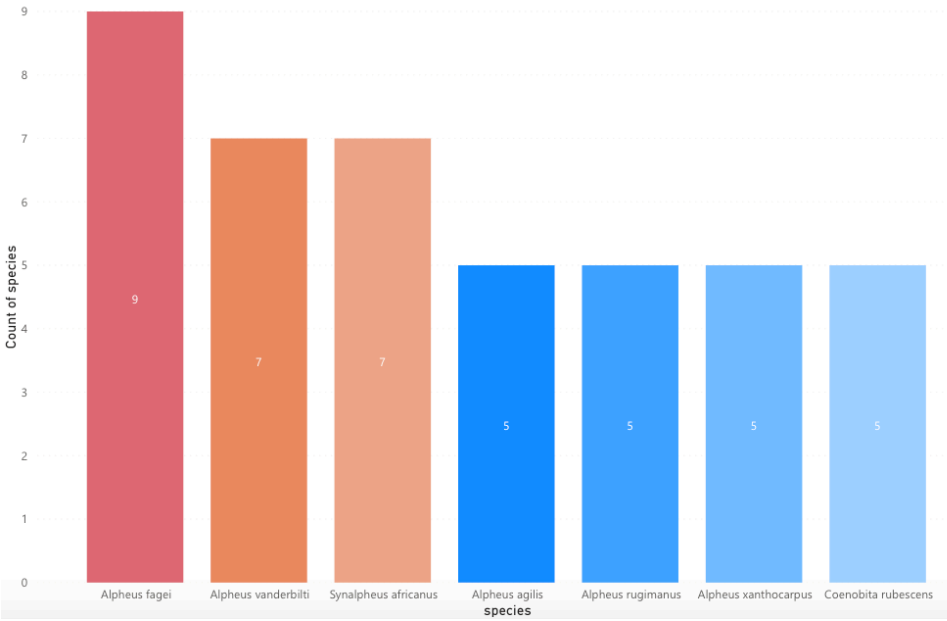

***Figure 3.9** – Top 7 most represented species occurrences in São Tomé and Príncipe with greater than or equal to 5 occurrences.*

Considering sheer numbers, i.e. number of individuals per occurrence, *Austruca annulipes* with 6488 individuals was the most representative species, followed by *Penaeus indicus* with 3010 individuals. A large number of species occurrences were represented by only one individual, among them *Zenopontonia rex* and *Xenophthalmodes moebii*. In MOZ *A. annulipes* and *P. indicus* presented the greatest number of individuals as well. See Figure 3.10.

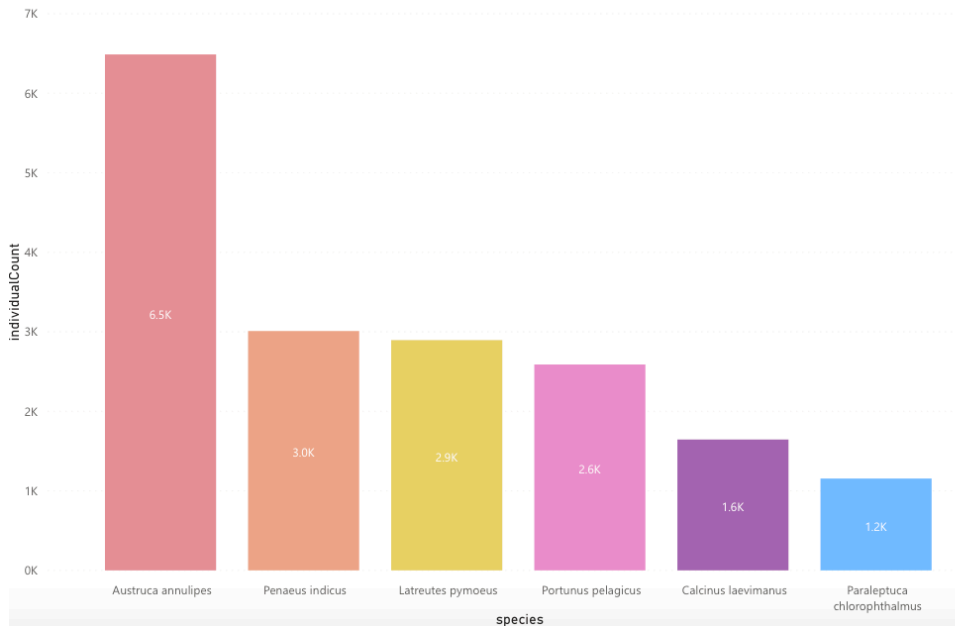

Figure 3.10 *– Top 6 most represented species by individual numbers in Mozambique and São Tomé and Príncipe with greater than or equal to 1000.*

In STP, the most representative species were *Deosergestes henseni* and *Sergestes henseni* with 75 individuals. A large number of species was represented by only one occurrence, among them *Troglocarcinus corallicola* and *Systellaspis debilis* species. See Figure 3.11.

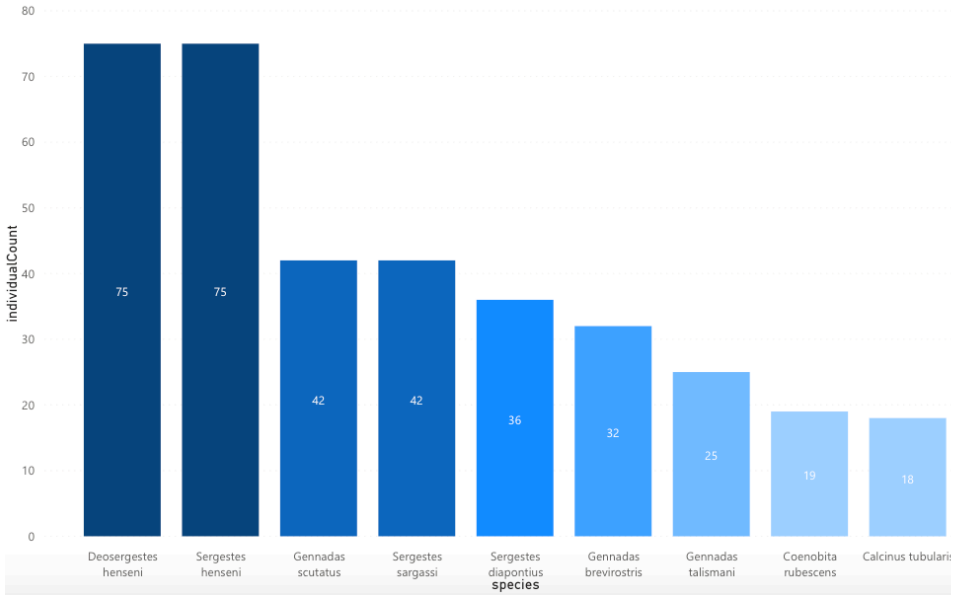

Figure 3.11 *– Top 9 most represented species by individual numbers in Mozambique and São Tomé and Príncipe with greater than or equal to 15.*

3.2. Data Management

Data was imported to the PostgresSQL RDBMS. The species occurrences data was organized in five Tables: 1) occurrences, 2) species, 3) location, 4) source and 5) spatial_ref_sys. The Table 1-4 refers to all species occurrences in MOZ and STP. Table 5 refers to a standardized catalog of the spatial reference systems, used to transform the database into a geodatabase. The remaining tables refer to habitats and areas of MOZ and STP. See Figure 3.12.

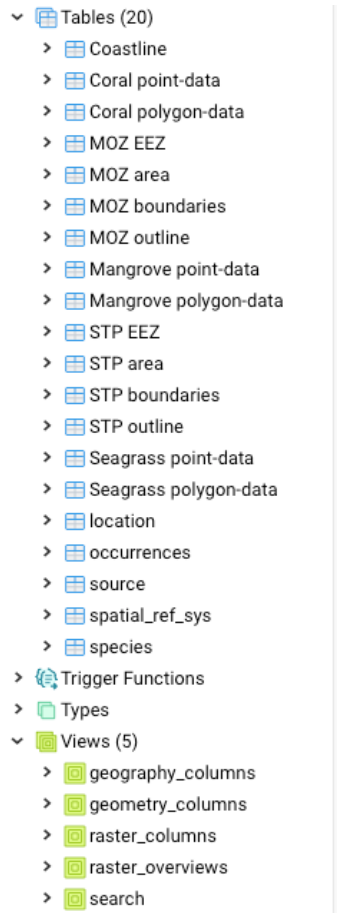

***Figure 3.12** – PostgreSQL RDBMS browser with the management tool pgAdmin.*

The location table has 9 columns corresponding to the DwC attributes catalogNumber, country, locality, verbatimLocality, habitat, decimalLatitude, decimalLongitude and geodeticDatum, plus one more column referring to latitude and longitude coordinates in geometryhexadecimal values, named coordinates. The occurrences table has 5 columns corresponding to the DwC attributes catalogNumber, depthInMeters, individualCount and eventDate. The source table has 4 columns corresponding to the DwC attributes catalogNumber, institutionCode, occurrenceID and references. The species table has 14 columns corresponding to the DwC attributes catalogNumber, scientificName, kingdom, phylum, class, order, family, genus, species, infraspecificepithet, taxonRank, previousName, taxonInfo and type.

The data from the database can be fetched by querying tables *(*e.g. species, country, habitat, etc.) using SELECT statements, for example:

```
SELECT species.scientificname, location.country, location.habitat
FROM species
INNER JOIN location ON species.catalogNumber=location.catalogNumber;
```

These statements were used to retrieve the data from the database tables, which returns data in a result table. See Figure 3.13.

	scientificname text	country text	habitat text
241	Aristeus antennat...	Mozambi...	Seagrass
242	Aristeus antennat...	Mozambi...	Coral Reef
243	Aristeus antennat...	Mozambi...	Seagrass
244	Aristeus antennat...	Mozambi...	Mangrove
245	Aristeus virilis (Sp...	Mozambi...	Open Sea
246	Aristeus virilis (Sp...	Mozambi...	Coastal Area
247	Aristeus virilis (Sp...	Mozambi...	Coastal Area
248	Aristeus virilis (Sp...	Mozambi...	Coastal Area
249	Aristeus virilis (Sp...	Mozambi...	Coral Reef
250	Aristeus virilis (Sp...	Mozambi...	Coral Reef
251	Aristeus virilis (Sp...	Mozambi...	Open Sea
252	Aristeus virilis (Sp...	Mozambi...	Open Sea
253	Ascidiophilus cap...	Mozambi...	Coastal Area
254	Ascidiophilus cap...	Mozambi...	Coastal Area
255	Ascidiophilus cap...	Mozambi...	Seagrass
256	Ascidiophilus cap...	Mozambi...	Seagrass
257	Ashtoret lunaris (...	Mozambi...	Coral Reef
258	Ashtoret lunaris (...	Mozambi...	Coral Reef
259	Ashtoret lunaris (...	Mozambi...	Coral Reef

***Figure 3.13** – Part of result-set from a SELECT query.*

The data can also be visualized using the PostGIS geometry viewer. It displays the graphical output of a geometry data column on *OpenStreetMap*. See Figure 3.14.

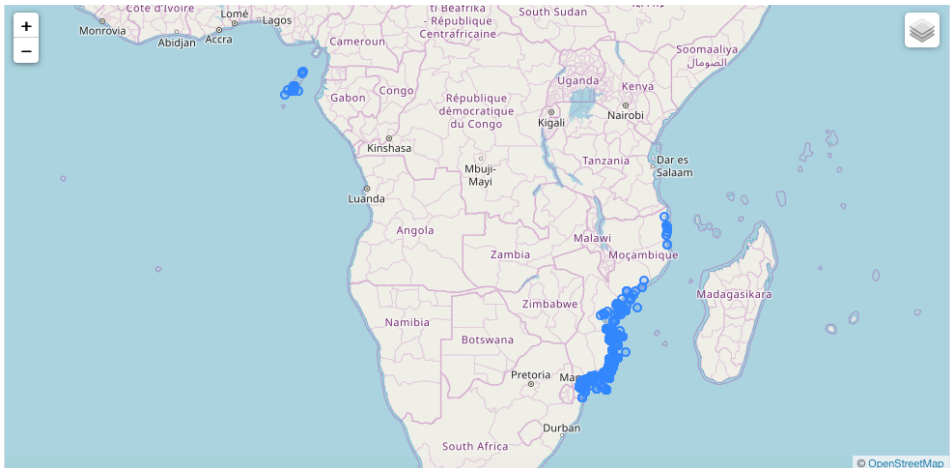

***Figure 3.14** – Graphical output of query on a geometry column displayed against an OpenStreetMap.*

3.3. Geographic Data Representation

This QGIS mapping layout, see Figure 3.15, was designed specifically for the COBIO-NET project, providing a basis for mapping species occurrence and distribution data across habitats in MOZ and STP (See Section 4.3.).

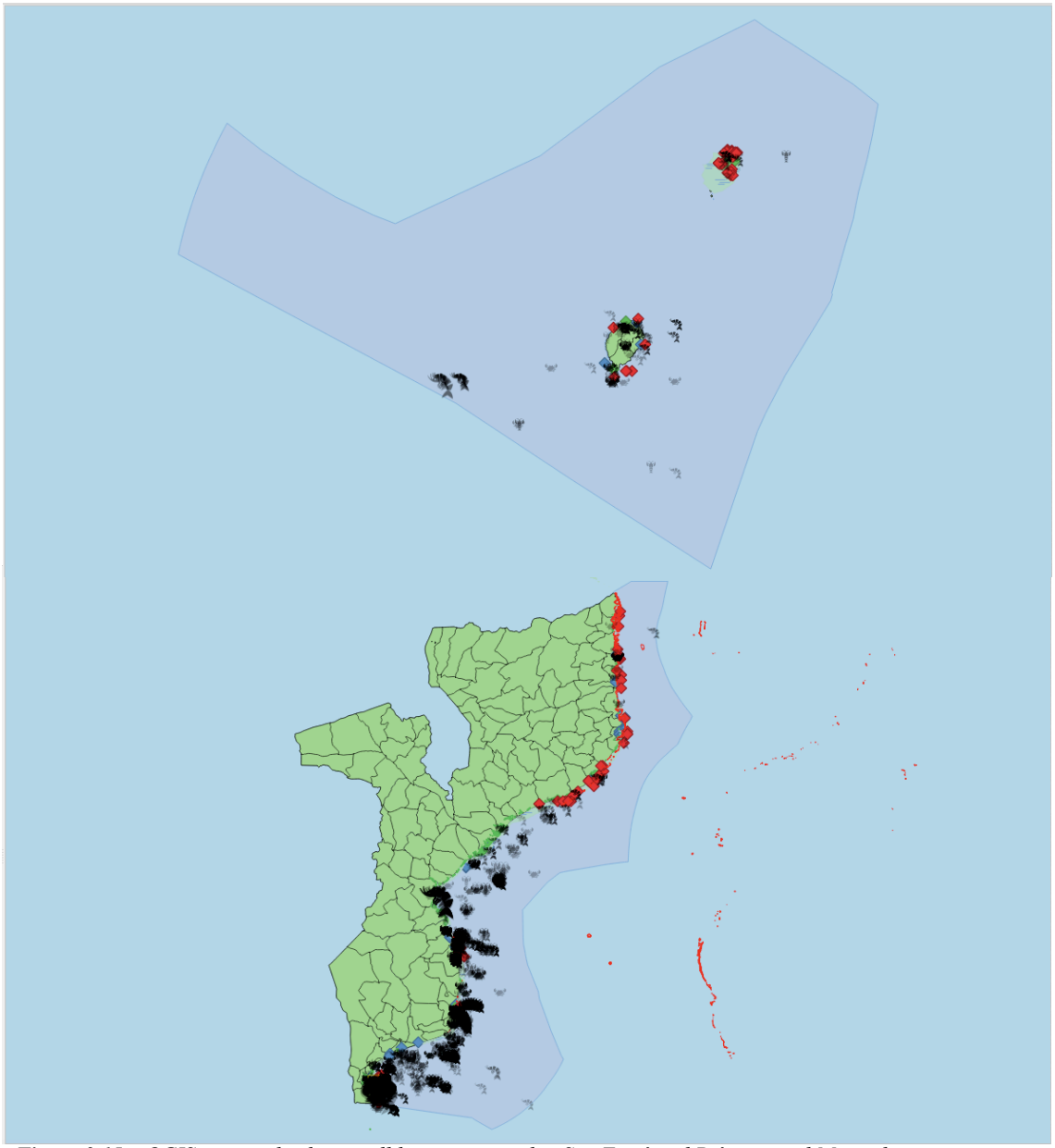

Figure 3.15 – *QGIS project displaying all layers assigned to São Tomé and Príncipe and Mozambique.*

3.4. Online Data Dissemination

The web map constructed here is an interactive digital platform, that can be used to visualize the occurrences of the decapod crustacean species along the coastlines of MOZ and STP. The map provides various filter layers to manipulate the data, allowing the visualization of occurrences against specific criteria (e.g. type of habitats, taxonomic classification, among others).

This web map contains 18 layers that can be selected or unselected to filter the data in display. These layers are grouped in 3 main sub-groups: 1) species occurrences, 2) habitats, and

3) MOZ and STP boundaries. To provide context, the global coastlines are also represented See Figure 3.16.

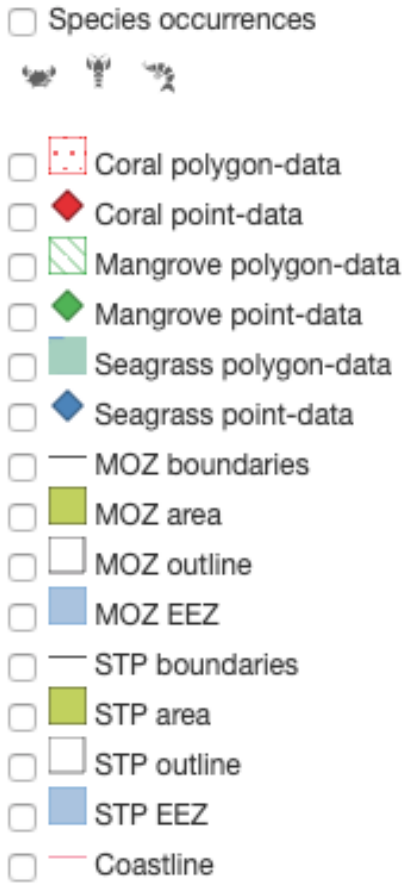

***Figure 3.16** – Web Map layers list menu.*

3.4.1. Species Occurrences

The species occurrences layers are separated into three main criteria: crab-like species, lobster-like species and shrimp-like species, each of them represented with an appropriate symbol. These icons can be seen varying in size, according to the number of individuals present by occurrence. An information box is shown by hovering each symbol with country, locality, habitat, latitude, longitude, depth, number of individuals, data, scientific name and taxonomy. See Figure 3.17.

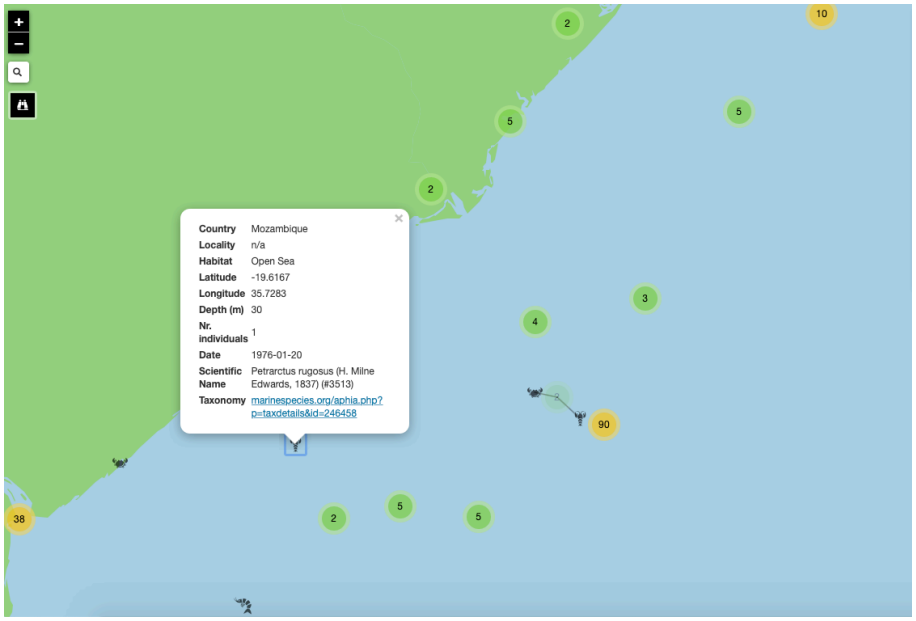

Figure 3.17 *– Web Map zoomed in Mozambique, showing the use of different symbols according with specimen typology.*

3.4.2. Habitats

The habitats are divided into three groups: corals, mangroves and seagrasses, with polygon-data and point-data layers, each represented with appropriate symbology. See Figure 3.18 and 3.19.

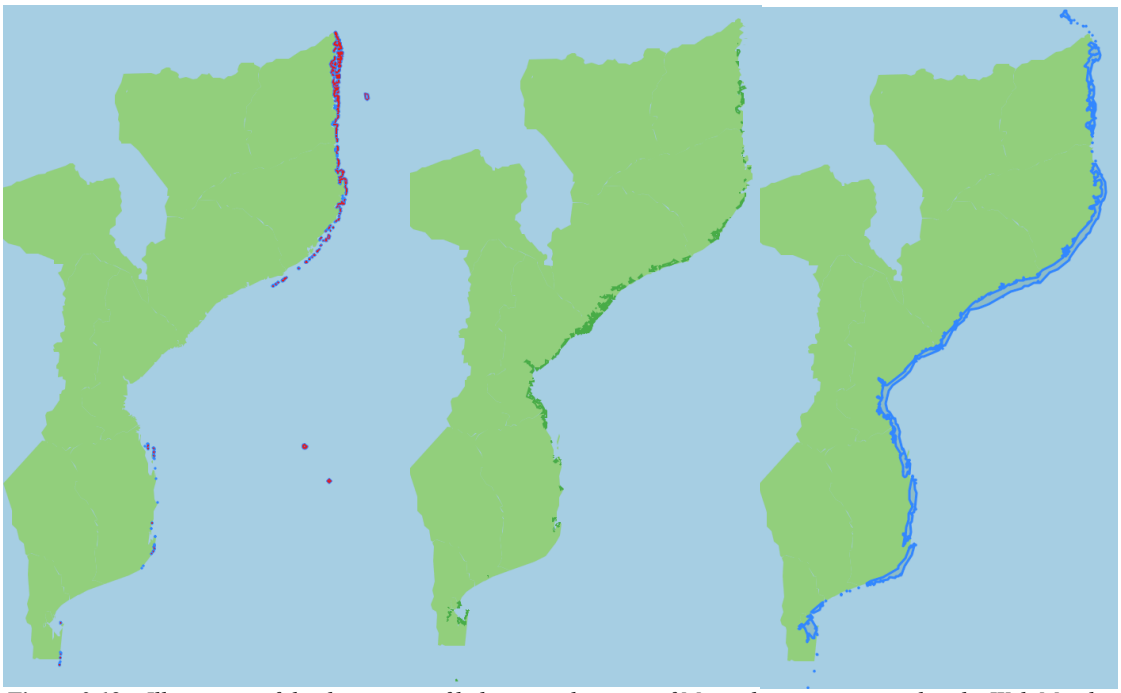

Figure 3.18 *– Illustration of the three types of habitats in the coast of Mozambique represented in the Web Map by polygon-data: coral, mangrove and seagrass, respectively.*

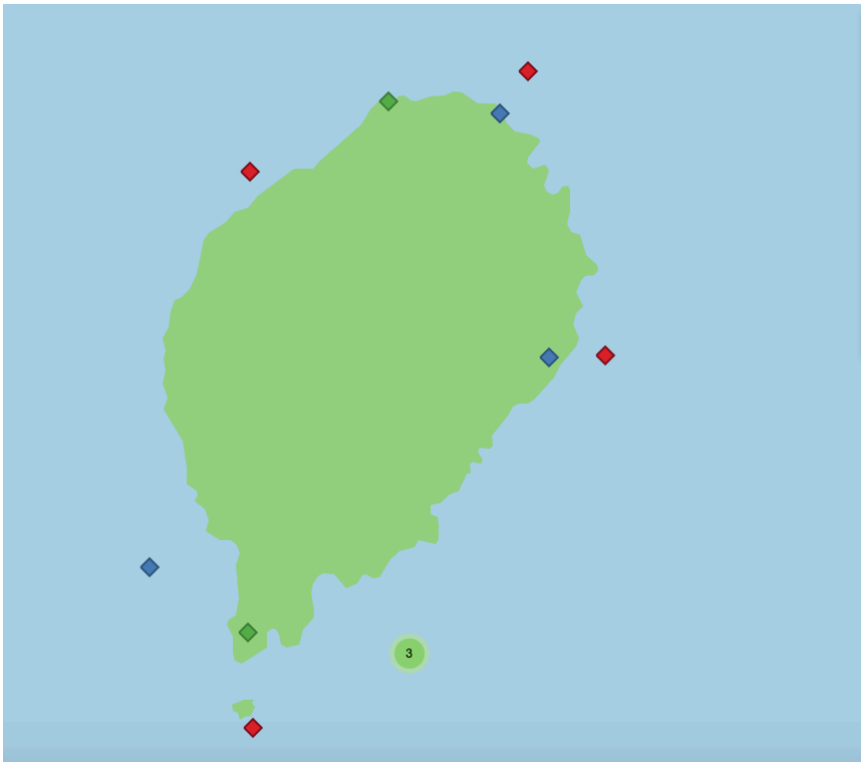

Figure 3.19 – *Illustration of the three types of habitats in the coast of São Tomé and Príncipe represented in the Web Map by point-data: coral in red, mangrove in green and seagrass in blue.*

3.4.3. Country Boundaries

Representation of STP and MOZ administrative boundaries, including their areas, as well as divisions of both MOZ and STP, outlining their borders and exclusive economic zones. See Figure 3.20.

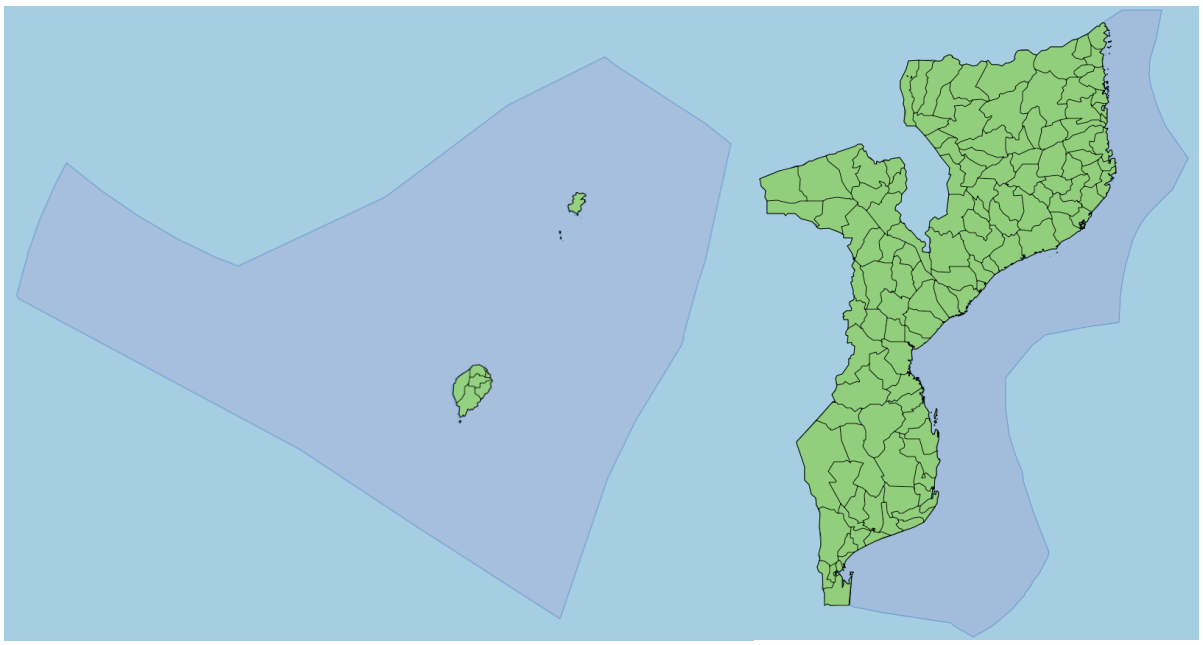

Figure 3.20 *– Representation of the boundaries of São Tomé and Príncipe (left) and Mozambique (right).*

3.4.4. *Appearance*

The web map is initialized in a full-screen template with a canvas presenting part of Africa, as well as the Atlantic and Indic oceans comprising the study areas. The layers menu is collapsed (top right side of the map) with four layers loaded (species occurrences, STP and MOZ areas and coastlines). The occurrences in the map are grouped in clusters, with three different categories; green colored circles range 1 to 10 items, yellow colored circles range 11 to 99 items and orange colored circles range more than 100 items. See Figure 3.21.

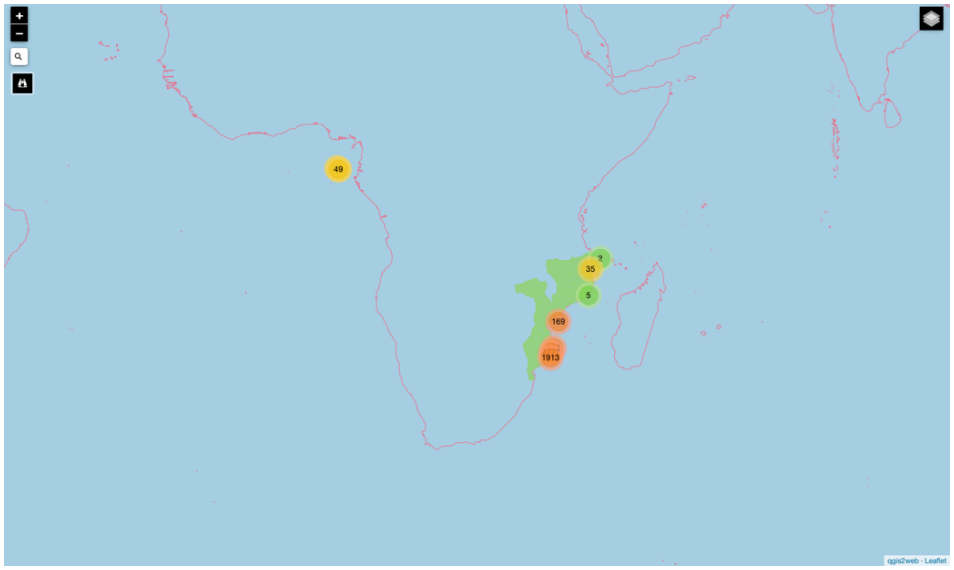

***Figure 3.21** – Web Map default appearance and scale/zoom.*

An address/locality search can be performed via the magnifying glass symbol. The search for species, genus or family occurrences can be accessed through the binocular symbol (top left side of the map). See Figure 3.22.

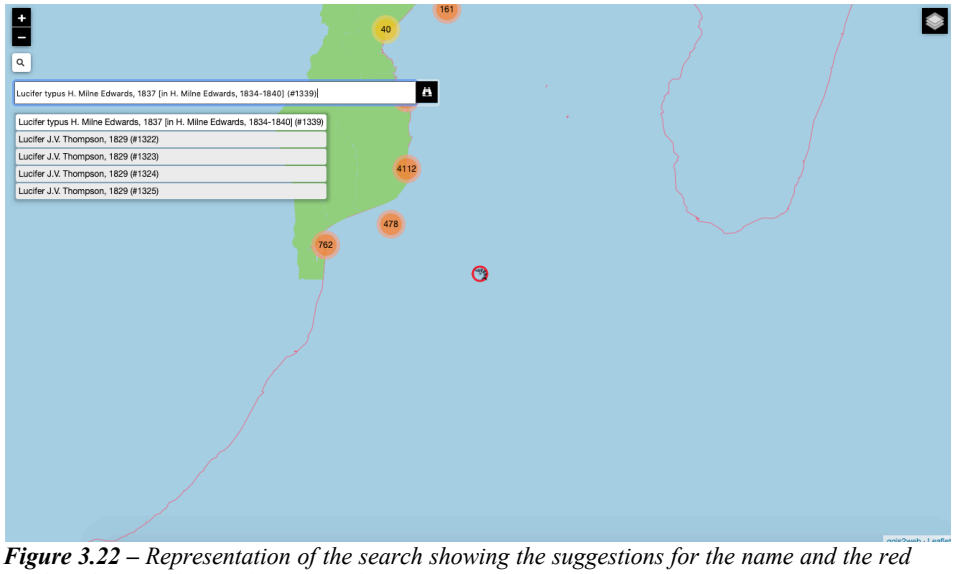

***Figure 3.22** – Representation of the search showing the suggestions for the name and the red circle indicating the result on the web map.*

3.4.5. Online Access

The interactive digital platform constructed can be accessed at the following link: https://cobio-net.000webhostapp.com/Web_Map/index.html

4. Discussion

Computer Science can be a great ally for biologists, providing them with tools for analysing and reporting findings on species and their behaviours. Due to the fact that there is a constant increase of the information available, which is mainly presented in digital format and only manageable through computer science techniques, ensuring its utility and accessibility is critical to current and future research in biological sciences (Torres *et al.*, 2006). However, there are a few challenges in what concerns handling large volumes of information, its integration from different sources and formats (heterogeneity), as well as the manipulation of data, images and geospatial reference information.

In this work, several digital tools were used to construct a dataset on decapod crustaceans and respective habitats through coastal zones in MOZ and STP. The complete and validated dataset was the basis to create maps within the COBIO-NET project, showing marine species occurrences in mangroves, seagrasses, corals, and other coastal areas of MOZ and STP. These maps were integrated into a web platform to assemble an interactive map on the distribution of marine decapod crustaceans across different habitats in MOZ and STP. The aim of this output is to share and disseminate information obtained during this study with the scientific community and the general public.

This section addresses the main achievements of this dissertation, the difficulties and limitations of the tools used, as well as possible alternative approaches to overcome these constraints. Further, it advances the application of the digital instruments developed here and then it discusses the integration of the outputs from the current dissertation into a wider project to accomplish long-term goals.

4.1. Biodiversity Dataset Construction: development of a dataset of MOZ and STP decapod crustaceans

In this study, information on decapod crustaceans obtained from different biodiversity sources (e.g. GBIF) was used as a pilot to develop a DwC structured dataset, with relevant information on species occurrences and ready to be analyzed for different purposes. To test the dataset several analyses were conducted, e.g. the most representative sources of information; the number of occurrences and distribution per country through different habitats; and the most represented families, species per country and habitat type. The significance of this biodiversity dataset is that it provides data for collecting and evaluating biodiversity information to, e.g., model species distributions and to estimate the size of species ranges in order to predict their risk of extinction and for advancing biodiversity conservation strategies, as suggested by Randrianasolo *et al.* (2002), Greenbaum and Komar (2005), and Newbold (2010).

The dataset constructed during this study is the digital support to store and manage information on marine invertebrates from MOZ and STP, within the COBIO-NET project. The dataset framework can be used by COBIO-NET researchers to assemble marine invertebrate data.

4.1.1. Limitations to the development of the dataset

Throughout this dissertation several tools and approaches were employed, which brought forth a number of challenges that had to be overcome.

One of these was, the cleaning up and validation of scientific names in the biodiversity dataset. The field of naming, circumscription and classification of species is in constant change, and thus it becomes difficult to grasp which of several published names correspond to our modern understanding of a specific species. The names

attached to species can be synonymized, moved between genera or even have their grammar corrected e.g. *Phymodius ungulatus* (H. Milne Edwards, 1834) accepted as *Cyclodius ungulatus* (H. Milne Edwards, 1834) (WoRMS, 2019).

There is also disagreement on higher ranks such as Family e.g. Calocarididae Ortmann, 1891 accepted as Axiidae Huxley, 1879 (WoRMS, 2019), or whether or not a particular rank is composed of two (or more) separated lineages, such as Thalassinidea now considered Axiidea and Gebiidea (Sakai, 2004; De Grave, 2009).

In order to validate scientific names, the use of taxonomic checklists is a must (all synonyms that have been used for a particular species are catalogued in checklists) since they evolve over time and serve as essential benchmarks in a constantly changing field. In this dissertation, WoRMS was used as the referenced taxonomic checklist to match scientific names. (WoRMS Editorial Board, 2019). Considering the one-year span of this study and that sometimes checklists contradict each other, it was imperative to use content which is up-to-date and controlled by taxonomic and thematic experts.

Digital repositories have new data entered daily, and often data is submitted in the form of large datasets, such as global or regional species occurrences lists. These large databases are not exempt from errors, inaccuracies and omissions, such as taxonomic uncertainties and geographical inaccuracies of species occurrences (Hortal *et al.*, 2015). This allows us to ponder, to what extent can these data be used in biodiversity or distribution studies to reveal general diversity patterns and identify main centers of diversity for a focal group.

In spite of these issues, digital repositories are quite useful, providing data uniformized from a number of sources that greatly exceed what could be gathered manually, saving time and money (Edwards, 2004; Chapman, 2005; Guralnick and Hill, 2009). But without *a priori* manual data cleaning and filling, incorrect taxonomic data, but most importantly, inaccurate geographic data, due to political subdivisions of the countries assigned after specimens collection, can lead to overestimation of species richness (Maldonado *et al.,* 2015). Therefore, an extensive amount of time was applied in cleaning up and verifying taxonomic and geographic data collected from the databases, always preserving the original data. GEOLocate was used because it is proved to be very useful for marine collections (Hill *et al.*, 2009). However, when this was not viable, uncertain data was removed (e.g. records with unknown locality not found either automatically by GEOLocate algorithm or manually with Google Maps like “CSB Estuário”, “Sulaca”, “Ingela Bank”; unknown species names like *Indopinnixa bifilia or Platylambus prensor*). The measure of uncertainty was not used on the web map, as species occurrences are rarely recorded precisely or precisely enough to be considered as true points, and therefore the uncertainty associated with the record means that the point actually represents an area (Wieczorek, 2001; Wieczorek *et al.*, 2004).

One might wonder if a peer-review requirement would solve any problems regarding accuracy in digital repositories, prior to making any occurrence data publicly available. However, considering the amount of work that had to be done and the many understudied species groups there are with only a handful of people in the world studying them, it wouldn’t be a realistic possibility to do so (Dov, 2007; Mora *et al.*, 2011). Though, a more practical solution can be found by allowing the participants and users of these open-source databases to provide feedback on specific records e.g. by correcting misidentifications and erroneous georeferencing data (Maldonado *et al.*, 2015).

4.1.2. *Different approaches to the used methodology*

When validating scientific names, an alternative to the Python script "previousName.py" using WoRMS service in this study, is the OpenRefine reconciliation service. An essential tool for biodiversity data and scientific name cleanup by matching names to external identifiers. See Figure 4.1.

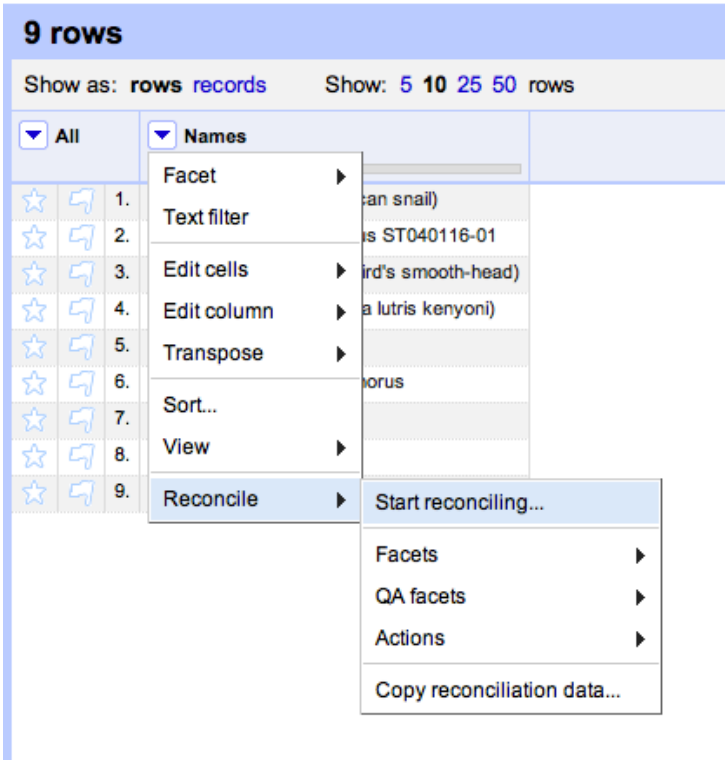

Figure 4.1 *– Reconciling function under column header → Reconcile → Start reconciling (Page, 2012).*

Quite a few services have been implemented throughout the years like Encyclopedia of Life (EoL), GBIF, Global Names Index (gni), NCBI Taxonomic, uBio FindIT, WoRMS, among others. These reconciliation services can identify one or more possible match names recognized in similar ways by different services (Delpeuch, 2019) and can also be used as another approach.

Showing the reconciling names linked to uBio, see Figure 4.2, one can create a column based on the reconciliation through the following command Edit column → Add column based on this column, with the expression:

4.1 cell.recon.match.name

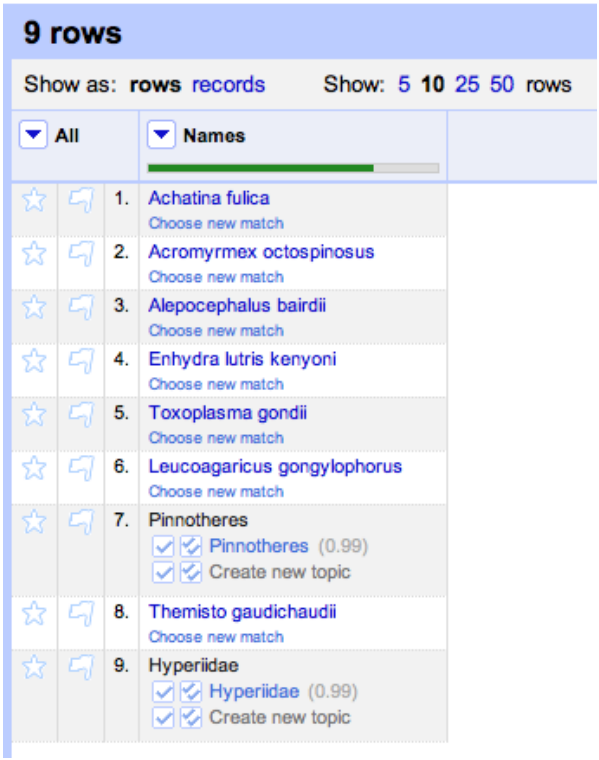

Figure 4.2 *– Result of reconciling names linked to uBio (Page, 2012).*

However, the functionality of these services and other automated workflows (Mathew *et al.*, 2014) are dependent of the website's API, which tend to decay over time, and at the time this dissertation was written, they were no longer available, because either the websites ceased to exist, were redesigned or the API was simply changed. Thus, although these methods can be used as alternative approaches, the approach using Python coding here reveled to be more adequate.

4.2. Data Management: using the PostgreSQL RDBMS

The dataset was exported to an open-source RDBMS, PostgreSQL, for management and processing. The database is organized by species occurrences, and it is the digital support to manage and process information on marine invertebrates from MOZ and STP, within the COBIO-NET project. Data from the database can be fetch through queries. This database is ready to be updated by researchers enrolled in COBIO-NET with more data from different marine invertebrate taxonomic groups.

4.2.1. Limitations of the PostgreSQL RDBMS

Several limitations have been associated with PostgreSQL, ranging from memory performance, in which it has a poorer performance in read-heavy operations when compared to, for example, MySQL, to not having as many accessible third-party tools to manage a database due to being less popular than MySQL. However, as for the spatial information manipulation functions that MySQL provides, they aren't as complete as the POSTGIS plugin (Fernandes, 2015).

Although these and other limitations do exist in PostgreSQL, no issues have directly affected this dissertation work. However, the particular error "date/time field

value out of range" was an issue, since PostgreSQL supports date only in "yyyy/mm/dd" format, as well as syntax violations such as column names are case-sensitive and have to be exclusively in lower case.

4.2.2. Different approaches to the used methodology

The use of PostgreSQL as the RDBMS for this project was chosen due to being open-source and its support to spatial type data. Other database options available in QGIS are Microsoft SQL Server (MSSQL), ESRI Personal GeoDatabase and Open Database Connectivity (ODBC). The MySQL, another widely used tool, should be an alternative RDBMS. However, the lack of support to MySQL was noticed, even though MySQL supports spatial data, QGIS does not support a Database connection to MySQL server since version 2.0.

4.3. Geographic Data Representation: mapping with QGIS

In this study, the manipulation and presentation of species occurrences and distribution was depicted by the open-source GIS application tool, QGIS. The software was used to view, edit and analyze the geographic data and information from the database, as well as setting up habitat layers, merging different habitat layers, adding background layers and changing symbology. The QGIS map was designed for the COBIO-NET project, providing a basis for mapping marine invertebrate species occurrences and distribution data across habitats in MOZ and STP. The QGIS map was the base map to web mapping using Leaflet.

4.3.1. Limitations of the shapefiles data

The main limitation found lies on the habitat shapefiles, that were compiled from multiple data sources with varying scale and quality, to which the image interpretation was conducted. Most of polygons align relatively well spatially to the coastline layer. However, these factors can produce a mismatch in the position of the layers in relation to the coastline. Other polygons were created using satellite imagery, and while some used a consistent methodology over all regions, others included observational data from various regional, national and international sources. Considering that the sources were different, the accuracy of the data may vary between locations and associated errors were not consistent across the datasets, including cloud cover, background noise, Landsat scanline error, certain areas misclassified due to striping artefacts, etc. While in many cases the datasets didn't undergo external review, some were validated by consulted experts with comprehensive field knowledge. Still, due to the higher presence of terrestrial vegetation, the accuracy is better determined in the seaward side compared to the landward side. The vectorization process, in some areas, produced many thousands of small polygons, which created overlapping polygons. To correct these, a dissolve operation with the software Mapshaper was performed (Spalding *et al.*, 2010; Giri *et al.*, 2011; UNEP-WCMC and Short FT., 2017; Bunting *et al.*, 2018; UNEP-WCMC *et al.*, 2018).

4.3.2. *Different approaches to the used methodology*

The habitat shapefiles concern a big enough time range to identify temporal shift patterns in their distribution. That turned out to be impossible, due to the fact that data from different years were aggregated, and not individually discriminated per year in the metadata. However, that might be possible if one researches further into the associated information.

4.4. Online Data Dissemination: web maps for geographic visualization

To promote online data dissemination, the development of an interactive web map to visualize the occurrences of decapod crustacean species along the coastlines of MOZ and STP coastlines, was accomplished using the open-source JavaScript library, Leaflet. The QGIS project map was used as a basis to build a web map using the qgis2web plugin for QGIS. In addition to the tools added in the basic application, other tools were added with Leaflet plugins. The map provides various filter layers to manipulate the data, allowing the visualization of occurrences against specific criteria.

4.4.1. *Limitations of the online data dissemination instruments*

Another limitation encountered in this dissertation, was the Leaflet web map generated from the QGIS map project with qgis2web plugin. Even though this plugin can emulate many aspects of the QGIS features, including symbology of the layers and styles like categorized or graduated, it cannot replicate more complex aspects.

One such aspect is the inability to symbolize using two variables for color and size. Another is the inclusion of grouped layers. Grouped layers are very poorly supported by qgis2web and the icons for the layers in the top-left panel can only be shown as individual layers but cannot be grouped. See Figure 4.3 showing the QGIS project layer menu depicting grouped layers in 6 groups: Occurrences, Coral, Mangrove, Seagrass, MOZ and STP.

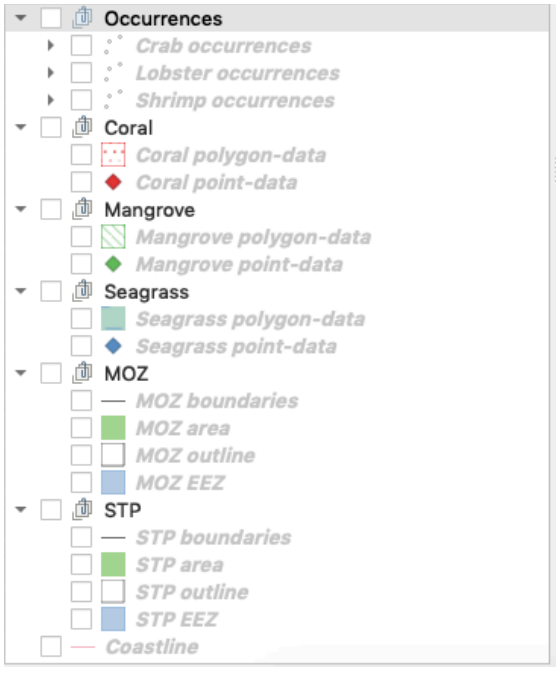

Figure 4.3 *– QGIS project layers menu depicting grouped layers in 6 groups: Occurrences, Coral, Mangrove, Seagrass, MOZ and STP*

The Leaflet Library cannot support the search for multiple items with the same name in one or more layers. For example, to search for a specific species in the scientific name attribute in the species occurrences layer, only one result and one result is shown, even though multiple occurrences results are possible. To make the search design work, the solution found was to assign a specific number to each line of all species occurrences in the search attribute that was previously created in the database as a view (search), allowing multiple search results to be displayed.

4.4.2. Different approaches to the used methodology

As one of the most popular open-source JavaScript libraries for interactive maps, Leaflet can also be used with other GIS software. An alternative to QGIS and qgis2web plugin is ESRI Leaflet, which is a collection of API plugins for using ArcGIS services and Leaflet applications. Leaflet also exists as a R package, one can integrate, control and create Leaflet maps in R console or RStudio.

The web service standard used to create this project was WMS, widespread to deliver map image in the format of raster and vector. WMS allows a standard interface to request for a geospatial map image and the available layers metadata were used to 1) produce a map and 2) simple querying of data. However, the representation created in WMS is only one static instance of the data i.e. the information exported with qgis2web is not updated whenever the database is changed. To feed the map with live data from the database on the web, the WFS service has to be used. A WFS service permits direct access to the server and its features allowing actions such as 1) query and retrieving the features of the dataset and 2) adding, deleting and updating features in a dataset (Owens, 2006).

Real-time web maps are frequently used in weather maps and traffic maps and are often animated. This method wouldn't necessarily suit the project at hand, at least not at such an early stage. Still, Carto software could be used for implementing dynamic web maps, utilizing the open-source library, Torque, which has been used to visualize large-scale datasets like human movement, Twitter user reactions to news and events, and biodiversity data as well e.g. LifeWatch Research Institute for Nature and Forest (INBO) project at https://inbo.carto.com/u/lifewatch (Alonso, 2017).

4.5. Applications of the Developed Digital Tools

Collection data and associated files can be shared through online digital repositories. Institutional repositories are essential for institutes to be able to control and submit the information they produce and manage. It is important that the information consulted in international databases can be linked to the home institution, directing the interested public in more specific information.

The DwC metadata structure developed in this dissertation is the basis to assemble a marine invertebrate biodiversity dataset for the COBIO-NET project, which will be used to manage marine invertebrate data produced by this project. The map generated in QGIS is ready for linking data from a larger dataset containing data from the other marine invertebrate taxonomic groups, to be added by the COBIO-NET investigators as long as it follows the structure of the dataset constructed in this study for decapod crustaceans. It can be used by researchers, for example, to visualize data, make distribution maps of either species or habitats or both, by using SQL expressions to query the data.

This can be achieved by using the DB Manager in QGIS, with the following path command:

Database → Database Manager → Virtual Layers → Project layers

To query these tables by clicking on the SQL Window (wrench icon) and write the desired SQL query e.g.:

```
Select * from "Species occurrences" where species_genus = 'Galathea'
```

The output is displayed in a SQL window as shown in Figure 4.4.

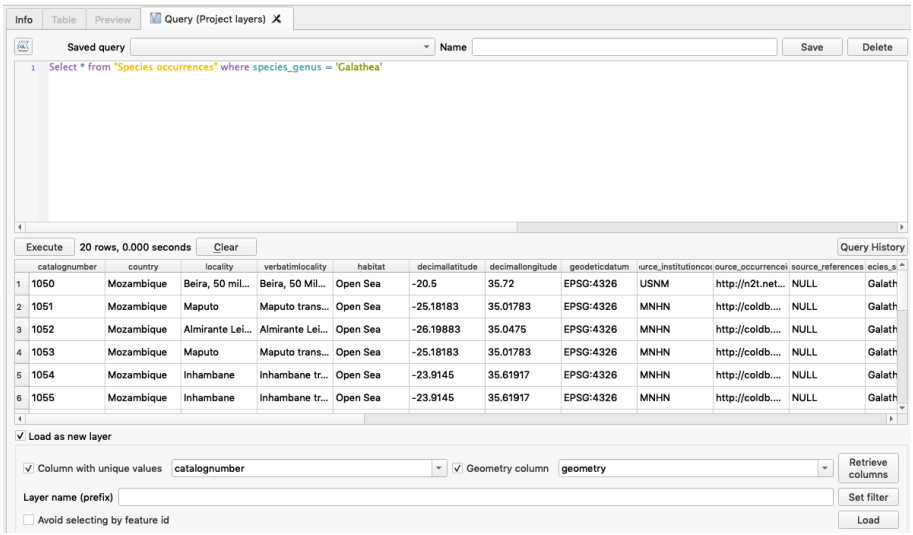

Figure 4.4 *– Query output displayed in a SQL Window.*

The output can then be loaded in the layer manager, as presented in Figure 4.5.

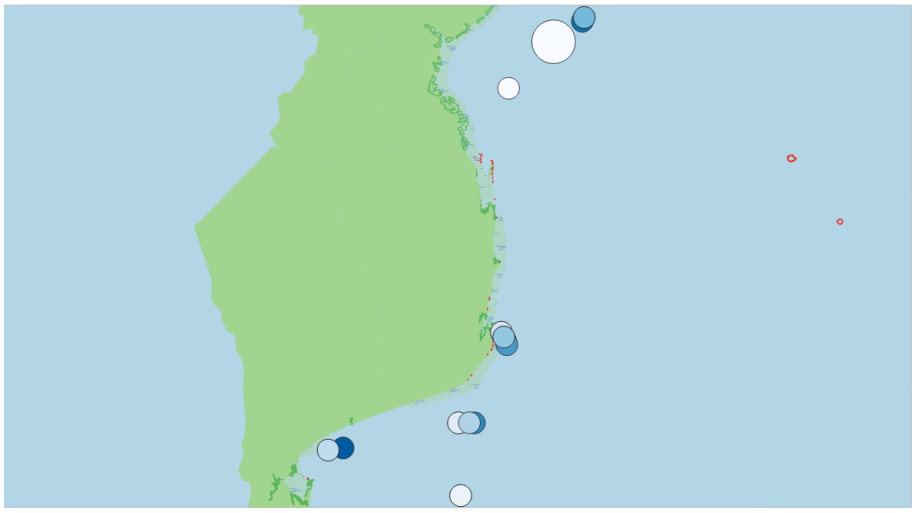

Figure 4.5 *– Map obtained; graduated symbology by method color and size defined by field individualCount; the size of the circles represents the individualCount field and the color represents the depthInMeters field (darkens with depth).*

The output can also be used to map animated maps using TimeManager plugin for QGIS (Friedrich, 2014; Nga *et al.*, 2012). This plugin exports several individual frames from a specific time interval and then combines them into an animation. The species layers can be used to map the species occurrences over time, using the eventDate field. See Figure 4.6 as an example to visualize the data species occurrences, through years 1921 to 2017.

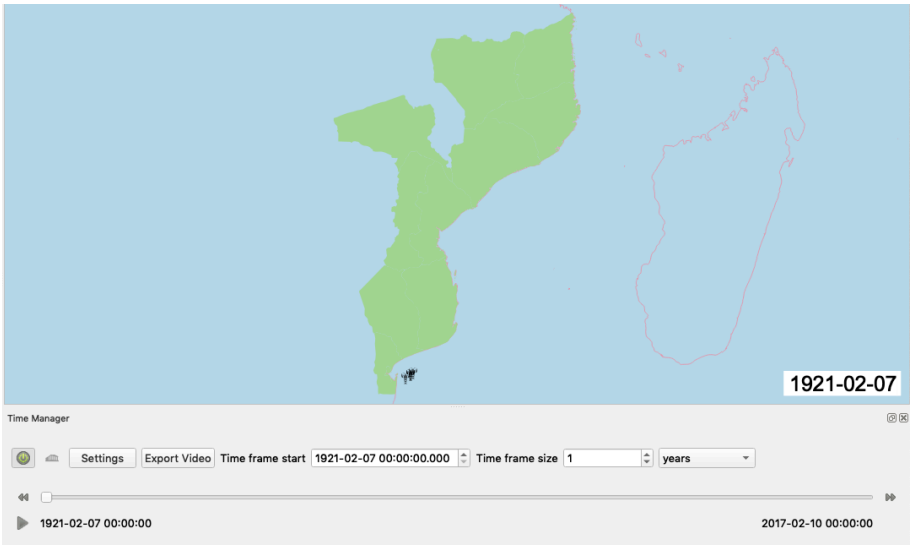

***Figure 4.6** – TimeManager toolbar in QGIS data window. The Settings button is used to access the layer selection option. The scroll buttons and slider enable navigation through the data species occurrences, through years 1921 to 2017.*

The animation can vary in a number of ways, such as time frame size from microseconds to years or accumulate features over time and allows the easy visualization of the data temporal dimension.

In summary, the tools developed (dataset, database, QGIS map project and web map) have several applications and will contribute to a better understanding of biodiversity databases with international standards (specifically DwC structured databases) and associated datasets enclosing comprehensive data. These digital appliances also permit the tracking of the original information, the discrimination of data from different sources (NHC, bibliography, checklists, digital repositories, etc.), mapping the information contained in biodiversity datasets (e.g. species occurrences and distribution across different habitat types) and the development of interactive maps, that can be easily accessed and used by the scientific community and the general public.

4.6. Long-term goals and perspectives

This dissertation is a part of the COBIO-NET project; Coastal biodiversity and food security in peri-urban Sub-Saharan Africa: assessment, capacity building and regional networking in contrasting Indian and Atlantic Ocean. This section of the COBIO-NET project developed a DwC structured dataset adequate to further log in marine invertebrate biodiversity data from MOZ and STP, across different coastal habitats, catalogued in different DwC structured databases, with the future purpose to make said information publicly available and accessible through their publication via global biodiversity repositories (e.g. GBIF) in both DwC and Ecological Metadata Language (EML) formats (Michener *et al.*, 1997; Olson and McCord, 2000). The final dataset will comprise a reference biodiversity collection encompassing the available information on marine invertebrates from MOZ and STP, contained in worldwide NHC, bibliographies and

checklists, to be accessed by the scientific community and the public through online consultation and use. The significance of the final dataset hinges not only in connecting information in a database for data management, but also, in connecting the maps produced in QGIS for potential use by the researchers of the project. The final output in the form of an interactive web map, will allow both researchers and the general public to visualize the information produced by the project.

One of the objectives of the COBIO-NET is to manage and centralize all datasets of biodiversity data produced during the project, and associated documentation as multimedia resources, in a digital repository. This repository will be used to disseminate the assembled information and data to the scientific community and the general public. Therefore, the dataset, QGIS maps and interactive maps produce during this dissertation will be included in the COBIO-NET digital repository for further dissemination and society outreach. The PostgreSQL database can also be used to develop a digitally adapted collection management structure, such as Specify at https://www.sustain.specifysoftware.org and Collective Access at https://www.collectiveaccess.org, that are free and open-source collection management software systems available. The first software, Specify, is NHC specific, while the Collective Access software is configured for any type of collection and both allow for the implementation of online platforms for data availability (Collectiveaccess.org, 2019; Specifysoftware.org, 2019).

Furthermore, the features of this dataset as a marine invertebrate biodiversity DwC structured database, the database itself, the maps and the web map, will be published in scientific articles, data papers and digital repositories (GBIF and COBIO-NET), within the scope of the COBIO-NET in collaboration with the other researchers of the project. Additionally, by being publicly available, these data and digital instruments, may be further handled by other users as teaching or dissemination tools, to write books, scientific articles, outreach brochures, etc.

The data gathered during the present work, the digital tools developed and the following publication of all these outputs further adds and promotes our current knowledge on marine invertebrate biodiversity in mangroves, seagrasses, corals and other coastal areas of MOZ and STP. This information is of importance, as it can be related to the natural and gastronomical resources of these two countries and the United Nations (UN) sustainable development goals (SDGs). In fact, one of the main aims of the COBIO-NET is to contribute to food security, a part of the SDG #2: “No hunger”, aiming to end hunger, achieve food security and improved nutrition worldwide, and marine invertebrates are relevant components of the diet and livelihoods of MOZ and STP populations (United Nations Development Programme, 2019). The reported information on marine invertebrates will also contribute to the SDG #14: “Life below water” which refers to marine and coastal biodiversity, its conservation and sustainable use for a sustainable development of the human society (United Nations Development Programme, 2019).

5. References

Agafonkin, V. (2019). *Leaflet Overview.* Available from https://leafletjs.com/ [14 August 2019].

Almeida, A. O. De, Coelho, P. A., Luz, J. R., Santos, Tiago, J. & Ferraz, N. R. (2008). Decapod crustaceans in fresh waters of southeastern Bahia, Brazil. *Revista de Biologia Tropical*, 56, 1225-1254.

Alonso, P. (2017). *Torque*. Available from https://github.com/cartodb/torque/ [27 August].

Baker, R., Phillips, C., Bradley, R., Burns, J., Cooke, D., Edson, G., Haragan, D., Jones, C., Monk, R., Montford, J., Schmidly, D.J. & Parker, N. (1998). Bioinformatics, museums and society: Integrating biological data for knowledge-based decisions. *Occasional Papers, Museum of Texas Tech University* 187, 1-4. DOI: 10.5962/bhl.title.143164

Baskauf, S., Wieczorek, J., Deck, J. & Webb, C. (2016). Lessons learned from adapting the Darwin Core vocabulary standard for use in RDF. *Semantic Web* 7, 617-627. DOI: 10.3233/SW-150199

Beaman, R. & Conn, B. (2003). Automated geoparsing and georeferencing of Malesian collection locality data. *Telopea* 10, 43-52. DOI: 10.7751/telopea20035604

Bloom, T., Flower, A. & DeChaine, E. (2018). Why georeferencing matters: Introducing a practical protocol to prepare species occurrence records for spatial analysis. *Ecology and Evolution 8*, 765–777. DOI: 10.1002/ece3. 3516

Brisby, F. (2000). The Quiet Revolution: Biodiversity Informatics and the Internet. *Science* 289, 2309-2312. DOI: 10.1126/science.289.5488.2309

Bunting, P., Rosenqvist, A., Lucas, R., Rebelo, L-M., Hilarides, L., Thomas, N., Hardy, A., Itoh, T., Shimada, M. & Finlayson, C. M. (2018). The Global Mangrove Watch – a New 2010 Global Baseline of Mangrove Extent. *Remote Sensing*, *2018* [under review].

Burke, L., Reytar, K., Spalding, M. & Perry, A. (2011). *Reefs At Risk Revisited*. World Resources Institute*,* Washington, DC: 130.

Câmara, G., Casanova, M. A., Magalhães, G. C. & Medeiros, C. M. B. (1996). Anatomia de Sistemas de Informação Geográfica. *10ª Escola de Computação, Instituto de Computação, Unicamp, Campinas*: 193.

Cartaxana, A., Correia, A. M., Carvalho, D. & Lopes, L. (2014). O papel das coleções de história natural no estudo e conservação de invertebrados. *Ecologi@*, 7, 15-21.

Chapman, A. (1991). Quality Control and Validation of Environmental Resource Data. *In: Data Quality and Standards Seminar.* Commonwealth Land Information Forum, Canberra: 1-12.

Chapman, A. (2005). Principles and Methods of Data Cleaning: Primary Species and Species-Occurrence Data, version 1.0. Report for the Global Biodiversity Information Facility, Copenhagen: 72. Available online at http://www.gbif.org/document/80528

Chapman, A. & Milne, D. (1998). The Impact of Global Warming on the Distribution of Selected Australian Plant and Animal Species in relation to Soils and Vegetation. Environment Australia. Projects: Species Distribution Modelling and Climate Change & Climate Change Impacts, 146.

Chapman, A. & Wieczorek, J. (Eds.). 2006. *Guide to Best Practices for Georeferencing.* Global Biodiversity Information Facility, Copenhagen: 90. Available online at http://www.gbif.org/orc/?doc_id=1288

COGE.GEO-Locate.org (2019). Collaborative Georeferencing Data Portal for GEOLocate. Available from http://coge.geo-locate.org/ [31 January 2019].

Collectiveaccess.org (2019). About CollectiveAccess. Available from https://www.collectiveaccess.org/about/ [2 September 2019].

Cristofori, E. I., Balbo, S., Camaro W., Pasquali, P., Boccardo, P. & Demarchi, A. (2015). Flood risk web-mapping for decision makers: A service proposal based on satellite-derived precipitation analysis and geonode. *In: IEEE International Geoscience and Remote Sensing Symposium*, Milan: 1389-1392. DOI: 10.1109/IGARSS. 2015.7326036

Cristofori, E. I., Facello, A., Demarchi, A., Camaro, W., Fascendini, M., & Villanucci, A. (2016b). A geographic information system as support to the healthcare services of nomadic community, the Filtu Woreda case study. *In: Symposium on climate change adaptation in Africa 2016,* Addis Ababa. DOI: 10.1007/978-3-319-49520-0_6

Cristofori, E. I., Facello, A. & Demarchi, A. (2017). Visualize and Communicate Extreme Weather Risk to Improve Urban Resilience in Malawi. *In: Green Energy and Technology. Springer,* 133-150. DOI: 10.1007/978-3-319-59096-7_7 7.1

Darwin Core maintenance group, Biodiversity Information Standards (TDWG) (2014). Darwin Core. Zenodo. DOI: 10.5281/zenodo.592792

De Grave, S., Pentcheff, N. D., Ahyong, S. T. *et al.* (2009). A classification of living and fossil genera of decapod crustaceans. *Raffles Bulletin of Zoology*, *21*, 1–109.

Delpeuch, A. (2019). Reconcilable Data Sources. Revision July 23. Available from https://github.com/OpenRefine/OpenRefine/wiki/Reconcilable-Data-Sources/ [27 August].

Dov, F. (2007). A 'taxonomic affidavit': why it is needed? *Integrative Zoology*, 2, 57–59. DOI: 10.1111/j.1749-4877.2007.00044.

Edwards, J. L. (2004). Research and societal benefits of the Global Biodiversity Information Facility. *BioScience*, 54, 485-486. DOI: 10.1641/0006-3568(2004)054[0486:RASBOT]2.0.CO;2

Emmerson, W. D. (2016). *A Guide to, and Checklist for, the Decapoda of Namibia, South Africa and Mozambique (Volume 3).* Cambridge Scholars Publishing, 720.

Fernandes, T. (2015). GeoBiblio - Base de Dados Bibliográfica Georreferenciada. Dissertação de Mestrado em Engenharia Informática e de Computadores. Instituto Superior Técnico, Lisboa: 54.

Francis, J. & Torell, E. (2004). Human dimensions of coastal management in the Western Indian Ocean. *Ocean & Coastal Management,* 47, 299-307. DOI: 10.1080/08920753.2011.589207

Freire, J. & Gonzalez-Gurriaran, E. (1998). New approaches to the behavioural ecology of decapod crustaceans using telemetry and electronic tags. *Hydrobiologia,* 371-372. DOI: 10.1023/A:1017066920151

Friedrich, C. (2014). Comparison of ArcGIS and QGIS for applications in sustainable spatial planning. *Master's dissertation, Universität Wien, Wien:* 132.

Gandhi, U. (2019). Web Mapping with QGIS2Web. Available from https://www.qgistutorials.com/en/docs/web_mapping_with_qgis2web.html [25 July 2019].

GBIF.org (2018). GBIF Home Page. Available from https://www.gbif.org/ [21 September 2018].

GBIF Secretariat (2017). GBIF Backbone Taxonomy. Checklist dataset DOI: 10.15468/39omei/ accessed via GBIF.org on 2019-09-12.

Geo-locate.org (2019). GeoLocate Developer Resources. Available from https://www.geo-locate.org/developers/default.html/ [14 August 2019].

Giri, C., Ochieng, E., Tieszen, L. L. *et al.* (2011). Status and distribution of mangrove forests of the world using earth observation satellite data. *Global Ecology and Biogeography*, 20, 154-159. (version 1.3, updated by UNEP-WCMC). DOI: 10.1111/j.1466-8238.2010.00584.x; http://data.unep-wcmc.org/datasets/4/

Graham, C. H., Ferrier, S., Huettman, F., Moritz, C. & Peterson, A. T. (2004). New developments in museum-based informatics and applications in biodiversity analysis. *Trends in Ecology & Evolution* 19, 497-503. DOI: 10.1016/j.tree.2004.07.006

Greenbaum, E. & Komar, O. (2005). Threat assessment and conservation prioritization of the herpetofauna of El Salvador. *Biodiversity and Conservation* 14, 2377-95.

Guralnick, R. & Hill, A. (2009). Biodiversity informatics: automated approaches for documenting global biodiversity patterns and processes. *Bioinformatics* 25, 421-428. DOI: 10.1093/bioinformatics/btn659

Heck, S., Béné, C. & Reyes-Gaskin, R. (2007). Investing in African fisheries: building links to Millennium Development Goals. *Fish* 8, 211-226. DOI: 10.1111/j.1467-2679.2007.00251.x

Henriques, P. (2016). Desenvolvimento de uma solução WebSIG OpenSource - "Autarquia Livre". Dissertação de Mestrado em Ciências e Sistemas de Informação Geográfica. Nova Information Management School, Universidade Nova, Lisboa: 74.

Hill, A. W., Guralnick, R., Flemons, P., Beaman, R., Wieczorek, J., Chavan, A. R. V. & Remsen, D. (2009). Location, location, location: utilizing pipelines and services to more effectively georeference the world's biodiversity data. *BMC Bioinformatics 2009* 10 (Suppl. 14), S3. DOI: 10.1186/1471-2105-10-S14-S3

Hooge, P. & Taggart, S. (1998). Pacific halibut in Glacier Bay National Park, Alaska. *In*: Mac, M.J., P.A. Opler, C.E. Puckett Haecker and P.D. Doran (eds.) *Status and Trends of the Nation's Biological Resources*. Vol. 2. U.S. Department of the Interior, United States Geological Survey, Reston, 712-714.

Hooge, P., Eichenlaub, W. & Solomon, E. (1999). Using GIS to Analyse Animal Movements in the Marine Environment. *Unites States Geological Survey, Anchorage:* 37-51. DOI: 10.4027/spmmp.2001

Hortal, J., de Bello, F., Diniz-Filho, *et al*. (2015). Seven Shortfalls that Beset Large-Scale Knowledge of Biodiversity. *Annual Review of Ecology, Evolution, and Systematics* 46, 523-549. DOI: 10.1146/annurev-ecolsys-112414-054400

Hutchinson, J., Spalding, M. & zu Emgassen, P. (2014). The Role of Mangroves Fisheries Enhancement. *The Nature Conservancy and Wetlands International,* 54.

Lane, M. (1996). Roles of Natural History Collections. *Annals of the Missouri Botanical Garden* 83, 536-545.

Maldonado, C., Molina, C. I., Zizka, A. *et al.* (2015). Estimating species diversity and distribution in the era of Big Data: To what extent can we trust public databases? *Global Ecology and Biogeography* 24, 973-984. DOI: 10.1111/geb.12326

Mantelatto, F. L. M. & Sousa, L.M. (2000). Population dynamics of the hermit crab Paguristes tortugae Schmitt, 1933 (Anomura, Diogenidae) from Anchieta Island, Ubatuba, Brazil. *Nauplius* 8, 185-193.

Marinoni, L. & Peixoto, A. (2010). As coleções biológicas como fonte dinâmica e permanente de conhecimento sobre a biodiversidade. *Ciência e Cultura,* 62, 54-57.

Mathew, C., Güntsch, A., Obst, M., *et al.* (2014). A semi-automated workflow for biodiversity data retrieval, cleaning, and quality control. *Biodiversity Data Journal,* 2, e4221. DOI: 10.3897/BDJ.2.e4221

Matos, J. (2013). Criação do interface de um sistema de gestão museológico de história natural. Dissertação de Mestrado em Design de Comunicação. Escola Superior de Artes e Design, Matosinhos: 141.

Michener, W. K., James, W. B., Helly, J. *et al*. (1997). Non-geospatial metadata for the ecological sciences. *Ecological Applications*, 7, 330-342. DOI: 10.2307/2269427

Mora, C., Tittensor, D. P., Adl, S., Simpson, A. G. B. & Worm, B. (2011). How many species are there on Earth and in the ocean? *PLoS Biology*, 9, e1001127. DOI: 10.1371/journal.pbio.1001127

Murphey, P. C., Guralnick, R. P., Glaubitz, R., Neufeld, D. & Ryan, J. A. (2004). Georeferencing of museum collections: A review of problems and automated tools, and the methodology developed by the mountain and plains spatio-temporal database-informatics initiative (Mapstedi). *PhyloInformatics*, *3*, 1-29. DOI: http://doi.org/10.5281/zenodo.59792

Newbold, T. (2010) Applications and limitations of museum data for conservation and ecology, with particular attention to species distribution models. *Progress in Physical Geography,* 34, 3. DOI: DOI: 10.1177/ 0309133309355630

Ng, P. K. L., Guinot, D. & Davie P. J. F. (2008). Systema Brachyurorum: Part I. an annotated checklist of extant brachyuran crabs of the world. *The Raffles Bulletin of Zoology*, 17, 1-286.

Nga, D. V., See, O. H., Xuen, C. Y., & Chee, L. L. (2012). Visualization Techniques in Smart Grid. *Smart Grid and Renewable Energy*, 3, 175.

Olson, R. J. & McCord, R. A. (2000). Archiving ecological data and information. *In:* W. Michener and J. Brunt, (eds.) *Ecological data - design, management and processing. Blackwell Science, Malden,* 117-141.

Owens, B. (2006). The WFS and WMS Services. Available from http://blog.geoserver.org/2006/11/27/the-wfs-and-wms-services/ [27 August].

Page, R. (2012). Using Google Refine and taxonomic databases (EOL, NCBI, uBio, WORMS) to clean messy data. Available from http://iphylo.blogspot.com/2012/02/using-google-refine-and-taxonomic.html [27 August 2019].

Raes, N. & van 't Zelfde, M. (2016). Introduction to QGIS 2.14 - Mapping species distributions. Naturalis Biodiversity Center, Laiden: 68.

Ramos, L., Torres, A., Pulhin, F. & Lasco, R. (2012). Developing a georeferenced database of selected threatened forest tree species in the Philippines. *Philippine Journal of Science,* 141, 165- 177.

Randrianasolo, A., Miller, J. S. & Consiglio, T. K. (2002). Application of IUCN criteria and Red List categories to species of five Anacardiaceae genera in Madagascar. *Biodiversity and Conservation* 11, 1289–300.

Rios, N. E. & Bart, H. L. (2010). GEOLocate (Version 3.22) [Computer software]. Belle Chasse, Tulane University Museum of Natural History.

Robertson, T., Döring, M., Guralnick, R., Bloom, D., Wieczorek, J., Braak, K., Otegui, J., Russel, L., Desmet, P. (2014). The GBIF Integrated Publishing Toolkit: Facilitating the Efficient

Publishing of Biodiversity Data on the Internet. *Plos One. 9(8).* DOI: 10.1371/journal.pone.0102623

Rocha, L. A., Aleixo, A., Allen, G. *et al.* (2014). Specimen collection: An essential tool. *Science* 344: 814-815. DOI: 10.1126/science.344.6186.814

Sakai, K. (2004). The diphyletic nature of the Infraorder Thalassinidea (Decapoda, Pleocyemata) as derived from the morphology of the gastric mill. *Crustaceana*. 77, 1117-1129. DOI: 10.1163/1568540042900268

Santos, T. (2013). Implementação de uma Base de Dados Geográficos para a Gestão das Matérias-Primas do grupo Cimpor. Dissertação de Mestrado em Sistemas de Informação Geográfica, Tecnologias & Aplicações. Faculdade de Ciências da Universidade de Lisboa, Lisboa: 88.

Schindel, D. & Cook, J. (2018). The next generation of natural history collections. *PLoS Biology* 16, e2006125. DOI: 10.1371/journal.pbio.2006125/

Senterre B. & Wagner M. 2016. Standardization of data collection and creation of a biodiversity database: a PostgreSQL-PostGIS database for Island Conservation Society (Seychelles). Consultancy Report, Government of Seychelles, United Nations Development Programme, Victoria, Seychelles, 40.

Shanmughavel, P. (2007). An overview on biodiversity information in databases. *Views & Challenges, Bioinformation by Biomedical Informatics Publishing Group.* www.bioinformation.net

Silva, R. R., Sampaio, C. M. S. & Santos, J. A. (2004). Fecundity and fertility of Macrobrachium amazonicum (Crustacea, Palaemonidae). *Brazilian Journal of Biology*, 64, 489-500. DOI: 10.1590/S1519-69842004000300012

Soares, L. B. (2019). Georreferenciação Geolocate. COBIO-NET Task 2 Output – Annex 3.

Sourcepole AG (2018). QGIS Cloud Home. Available from https://qgiscloud.com [15 August 2019].

Spalding, M., Kainuma, M. & Collins, L. (2010). World Atlas of Mangroves (version 2.0). A collaborative project of ITTO, ISME, FAO, UNEP-WCMC, UNESCO-MAB, UNU-INWEH and TNC. Earthscan, London, 319. URL: http://data.unep-wcmc.org/datasets/5/

Specifysoftware.org (2019). About Specify. Available from https://www.sustain.specifysoftware.org/about/ [2 September 2019].

Suarez, A. V. & Tsutsui, N. D. (2004). The Value of Museum Collections for Research and Society. *Bioscience* 54, 66-74.

The PostgreSQL Global Development Group (2019). PostgreSQL 11.5 Documentation. Available from https://www.postgresql.org/docs/11/index.html/ [15 August 2019].

Torres, R. S., Medeiros, C. M. B., Calves, M. A. G. & Fox, E. A. (2006). A Digital Library Framework for Biodiversity Information System. *International Journal on Digital Libraries* 6, 3-17. DOI 10.1007/s00799-005-0124-1

UNEP-WCMC, WorldFish Centre, WRI & TNC (2018). Global distribution of coral reefs, compiled from multiple sources including the Millennium Coral Reef Mapping Project. Version 4.0, updated by UNEP-WCMC. Includes contributions from IMaRS- USF and IRD (2005), IMaRS-USF (2005) and Spalding *et al.* (2001). Cambridge (UK): UNEP World Conservation Monitoring Centre. URL: http://data.unep- wcmc.org/datasets/1

UNEP-WCMC & Short FT (2017). Global distribution of seagrasses (version 6.0). Sixth update to the data layer used in Green and Short (2003). Cambridge (UK): UN Environment World Conservation Monitoring Centre. URL: http://data.unep- wcmc.org/datasets/7

United Nations Development Programme (2019). Sustainable Development Goals – Goal 2: Zero Hunger. Available from https://www.un.org/sustainabledevelopment/hunger/ [7 September].

United Nations Development Programme (2019). Sustainable Development Goals – Goal 14: Life Below Water. Available from https://www.un.org/sustainabledevelopment/oceans/ [7 September].

Vincent, K., Roth, R. E., Moore, S. A., Huang, Q., Lally, N., Sack, C. M., Nost, E. & Rosenfeld, H. (2018). Improving spatial decision making using interactive maps: An empirical study on interface complexity and decision complexity in the North American hazardous waste trade. *Environment and Planning B: Urban Analytics and City Science, 18.* DOI: 10.1177/2399808318764122

Wahid, M. T., Sharif, A. R. M., Napis, S. & Ahmad, N. (2006). Spatio-Temporal Biodiversity Data Model Using Object Relational Modeling Approach. *IJCSNS International Journal of Computer Science and Network Security*, 6, 10.

White, G. & Garrott, R. (1990). *Analysis of Wildlife Radio-Tracking Data*. San Diego: Academic Press, 383.

Wieczorek, J. (2001). MaNIS/HerpNet/ORNIS Georeferencing Guidelines. Museum of Vertebrate Zoology, University of California, Berkley. Available: http://georeferencing.org/georefcalculator/docs/GeorefGuide.html

Wieczorek, J., Guo, Q., & Hijmans, R. (2004). The point-radius method for georeferencing locality descriptions and calculating associated uncertainty. *International Journal of Geographical Information Science*, *18*, 745–767.

Wieczorek, J., Bloom, D., Guralnick, R., Blum, S., Doring, M., Giovanni, R., Robertson, T. & Vieglais, D. (2012). Darwin Core: An Evolving Community-Developed Biodiversity Data Standard. PLoS ONE 7(1): e29715. DOI: 10.1371/journal.pone.0029715

Wieczorek, C., & Wieczorek, J. (2015). Georeferencing calculator (version 20151221). Museum of Vertebrate Zoology, University of California, Berkeley.

Worboys, M. & Duckham, M. (2004*). GIS: A Computing Perspective.* Florida: CRC Press, 448.

WoRMS (2019). Calocarididae Ortmann, 1891. Accessed at: http://www.marinespecies.org/aphia.php?p=taxdetails&id=106799 on 2019-08-20/

WoRMS (2019). *Phymodius ungulatus* (H. Milne Edwards, 1834). Accessed at: http://www.marinespecies.org/aphia.php?p=taxdetails&id=209129 on 2019-08-19/

WoRMS Editorial Board (2019). World Register of Marine Species. Available from http://www.marinespecies.org at VLIZ. Accessed 2019-08-20. doi:10.14284/170/

6. Annexes

6.1. Data Sources

Ahyong, S. (2014). Cymonomid crabs of the MAINBAZA Expedition (Decapoda: Brachyura). *Zootaxa* 3821, 90-384. DOI: 10.11646/zootaxa.3821.3.7.

Anker, A. (2017). Strongly carinate species of Alpheopsis Coutière, 1897 of the tropical Atlantic and eastern Pacific, with redescription of A. trigona (Rathbun, 1901) and description of three new species (Malacostraca: Decapoda: Alpheidae). Plazi.org taxonomic treatments database. Checklist dataset DOI: 10.11646/zootaxa.4277.2.2 accessed via GBIF.org on 2018-09-21.

Annals of the South African Museum. Annale van die Suid-Afrikaanse Museum. Cape Town: South African Museum, 1898-2004.

Arbizu, P. M., Smith, C. R., Keller, S. & Ebbe, B. (Editors). Biogeographic Database of the Census of Abyssal Marine Life. Available online at http://www.cedamar.org/biogeography, DOI: 10.15468/oc9tsb accessed via iOBIS on 2018-11-03.

Augusto A. V. Flores, João Saraiva, José Paula, Sexual Maturity, Reproductive Cycles, and Juvenile Recruitment of Perisesarma Guttatum (Brachyura, Sesarmidae) at Ponta Rasa Mangrove Swamp, Inhaca Island, Mozambique, *Journal of Crustacean Biology* 22, 143-156. DOI: 10.1163/20021975-99990217

Baba, K. & Tirmizi, N.M. (1979) A new chirostylid (Crustacea, Decapoda, Anomura) from deeper parts of the Japanese waters and off the east coast of Africa. *Proceedings of the Japanese Society of Systematic Zoology* 17, 52-57.

Baba, K. (2005). Deep-sea chirostylid and galatheid crustaceans (Decapoda: Anomura) from the Indo-West Pacific, with a list of species. Galathea Report 20, 1-317. DOI: 10.11646/zootaxa.4057.1.5

Berggren, M. (1990). Dasella herdmaniae (Lebour) (Decapoda: Natantia: Pontoniinae) from Moçambique and Establishment of a New Species, Dasella brucei. *Journal of Crustacean Biology* 10, 554-559. DOI:10.2307/1548347

Berggren, M. (1991). Athanopsis rubricinctuta, New Species (Decapoda: Natantia: Alpheidae), a Shrimp Associated with an Echiuroid at Inhaca Island, Mocambique. *Journal of Crustacean Biology* 11, 166-178. DOI: 10.2307/1548555.

Berggren, M. (1994). Periclimenes Nomadophila & Tulеariocaris Sarec, Two New Species of Pontoniine Shrimps (Decapoda: Pontoniinae), from Inhaca Island, Mozambique, *Journal of Crustacean Biology* 14, 782-802, DOI: 10.1163/193724094X00759.

Blum, S., Fong, J. (2016). CAS Invertebrate Zoology (IZ). Version 14.2. California Academy of Sciences. Occurrence dataset DOI: 10.15468/tiac99accessed via GBIF.org on 2018-09-21.

Bosman, A. (2017). FBIP:IZIKO-UCT:Historical Survey (1930-1980). South African National Biodiversity Institute. Occurrence dataset DOI: 10.15468/zmnk0m accessed via GBIF.org on 2019-09-21.

Bosman, A. (2018). FBIP:IZIKO-UCT:Historical Invertebrates (1930-1980). Version 1.1. South African National Biodiversity Institute. Occurrence dataset DOI: 10.15468/fk7bhk accessed via GBIF.org on 2018-09-21.

British Oceanographic Data Centre, UK. Biogeographic data from BODC. *In*: EurOBIS, http://www.marbef.org/data/eurobissearch.php?dataprovider=47accessed via iOBIS on 2018-11-03.

Brito, A. & Pena, A. (2007). Population structure and recruitment of penaeid shrimps from the Pungué River Estuary to the Sofala Bank Fishery, Mozambique. *Western Indian Ocean Journal of Marine Science* 6, 147-158.

Cabezas, P., Macpherson, E. & Machordom, A. (2010). Taxonomic revision of the genus Paramunida Baba, 1988 (Crustacea: Decapoda: Galatheidae): A morphological and molecular approach. *Zootaxa* 2712, 1-60. DOI: 10.11646/zootaxa.2712.1.1.

Cabezas, P, Macpherson, E. & Machordom, A. (2011). Allogalathea (Decapoda: Galatheidae): a monospecific genus of squat lobster? *Zoological Journal of the Linnean Society* 162, 245-270. DOI: 10.1111/j.1096-3642.2010.00681.x.

Castro, P., Williams, A. & Cooper, L. (2003). Revision of the family Latreilliidae Stimpson, 1858 (Crustacea, Decapoda, Brachyura). *Zoosystema*. 25.

Cauquil, P. & Barde, J. (2011). observe_tuna_bycatch_ecoscope. IRD - Institute of Research for Development. Occurrence dataset DOI: 10.15468/23m361 accessed via GBIF.org on 2018-10-03.

Calabuig, I. (2016). Galathea II, Danish Deep Sea Expedition 1950-52. Zoological Museum, Natural History Museum of Denmark. Occurrence dataset DOI: 10.15468/ouseij accessed via GBIF.org on 2018-09-21.

Cauquil, P. & Barde, J. (2011). observe_tuna_ecoscope. IRD - Institute of Research for Development. Occurrence dataset DOI: 10.15468/mkrtxh accessed via GBIF.org on 2018-10-03.

Chan, T., Cleva, R. & Chu, K. H. (2016). On the genus Trachysalambria Burkenroad, 1934 (Crustacea, Decapoda, Penaeidae), with descriptions of three new species. Plazi.org taxonomic treatments database. Checklist dataset DOI: 10.11646/zootaxa.4150.3.1 accessed via GBIF.org on 2018-09-21.

Chavan, VIshwas & Achuthankutty, C. T. (editors), IndOBIS Catalogue of Life, Available at http://www.indobis.org/ accessed on 2018-11-03.

Clark, P. F. & Galil, B. (1993). A revision of the xanthid genus Pilodius Dana, 1851 (Crustacea: Brachyura: Xanthoidea). *Journal of Natural History* 27. DOI: 10.1080/00222939300770691.

Clark, P. & Paula, J. (2003). Descriptions of ten xanthoidean (Crustacea: Decapoda: Brachyura) first stage zoeas from Inhaca Island, Mozambique. *Raffles Bulletin of Zoology* 51.

Creuwels, J. (2017). Naturalis Biodiversity Center (NL) - Museum collection digitized at storage unit level. Naturalis Biodiversity Center. Occurrence dataset DOI: 10.15468/17e8en accessed via GBIF.org on 2018-09-21.

Crosnier, A. & Forest, J. (1966). Résultats Scientifiques des Campagnes de la "Calypso". Fascicule VII. Campagne de la Calypso dans la Golfe de Guinée et aux Iles Principe, Sao Tomé et Annobon (1956), et campagne aux Iles du Cap Vert (1959) (suite). 19. Crustacés décapodes: Alpheidae. *Annales de l'Institut Océanographique* 44, 199-314.

de Freitas, A. J. (1985). The Penaeoidea of southeast Africa. II. The Families Aristeidae and Solenoceridae. *Oceanographic Research Institute, Investigational Report* 57, 1-69.

de Freitas, A. J. (1987) The Penaeoidea of southeast Africa. II. The Family Penaeidae (excluding genus Penaeus). *Oceanographic Research Institute, Investigational Report* 58, 1-104.

de Moor, F., Ranwashe, F. (2017). AM: Freshwater Invertebrates (1900-2005). Version 1.3. South African National Biodiversity Institute. Occurrence dataset DOI: 10.15468/spzgor accessed via GBIF.org on 2018-09-21.

Diveboard - Scuba diving citizen science observations. Online at http://www.diveboard.com and http://ipt.diveboard.com/resource.do?r=diveboard-occurrences. DOI: 10.15468/tnjrgy Balbaya database, managed by IRD, funded by DCF / DG-MARE accessed via iOBIS on 2018-11-03.

Estudos de Biologia Maritima. Ministério do Ultramar. Junta de Investigações do Ultramar. *Anais*, 1953, volume VIII, Tomo II - Fascículo I accessed via iOBIS on 2018-11-03.

European Nucleotide Archive (EMBL-EBI) (2019). Geographically tagged INSDC sequences. Occurrence dataset DOI: 10.15468/cndomv accessed via GBIF.org on 2018-09-21.

Fabri, M. C. *et al.*, Ifremer BIOCEAN database (Deep Sea Benthic Fauna). Institut Franais de Recherche pour l'Exploitation de la Mer, Ifremer, Issy-les-Moulineaux, France. World Wide Web electronic publication, http://www.ifremer.fr/isi/biocean accessed via iOBIS on 2018-11-03.

Farfante, I. P. & Ivanov, B. G. Mesopenaeus Mariae, a New Species of Shrimp (Penaeoidea: Solenoceridae), the First Record of the Genus in the Indo-west Pacific, *Journal of Crustacean Biology* 2(2), 1 April 1982, 303-313, DOI: 10.2307/1548011

Ferreira, L. A. D. A. & Tavares, M. (2017). A new species of Pachycheles (Crustacea: Anomura: Porcellanidae), with taxonomic remarks on two other porcelain crabs from the remote oceanic archipelago of Trindade and Martin Vaz, South Atlantic Ocean. Plazi.org taxonomic treatments database. Checklist DOI: 10.11646/zootaxa.4299.4.5 accessed via GBIF.org on 2018-09-21.

Forest, J. (1961) Pagurides de l'Afrique occidentale. Scientific Results of the Danish Expedition to the coasts of tropical West Africa 1945–1946. *Atlantide Report* 6, 203-250.

Forest, J. (1963). Sur deux Scyllarus de l'Atlantique tropical africain: S. paradoxus Miers et S. posteli sp. Nov. *Remarques sur les Scyllarus de l'Atlantique oriental, Bulletin de l'Institut Océanographique* (Monaco).

Fratini, S., Vannini, M., Cannicci, S. & Schubart, C. (2005). Tree-climbing mangrove crabs: A case of convergent evolution. *Evolutionary ecology research* 17, 219-233.

Galil, B. (2000). Crustacea Decapoda: Review of the genera and species of the family Polychelidae Wood-Mason, 1874. *In*: A. Crosnier (ed), Résultats des Campagnes MUSORSTOM, Volume 21. *Mémoires du Muséum national d'Histoire naturelle*, Paris, (Zoologie) 184, 285-387.

Galil, B. (2011). An examination of the genus Philyra Leach, 1817 (Crustacea, Decapoda, Leucosiidae) with descriptions of seven new genera and six new species. *Zoosystema* 31, 279-320. DOI: 10.5252/z2009n2a4.

Gall, L. (2019). Invertebrate Zoology Division, Yale Peabody Museum. Yale University Peabody Museum. Occurrence dataset DOI: 10.15468/0lkr3w accessed via GBIF.org on 2018-09-21.

García, E., Muñoz, I (2018). Colección de Crustáceos Decápodos y Estomatópodos del Centro Oceanográfico de Cádiz: CCDE-IEOCD. Version 1.7. Oceanographic Center of Cadiz, Spanish Institute of Oceanography (IEO). Occurrence dataset DOI: 10.15468/anztjy accessed via GBIF.org on 2018-09-21.

Guinot, D. & de Forges, B. R. (1985). Crustacés Décapodes: Majidae (genres Platymaia, Cyrtomaia, Pleistacantha, Sphenocarcinus et Naxioides). *I & II Phillipines, Tome 2 - Résultats des Campagnes Musorstom*.

Harvard University M, Morris, P. J. (2019). Museum of Comparative Zoology, Harvard University. Version 162.172. Museum of Comparative Zoology, Harvard University. Occurrence dataset DOI: 10.15468/p5rupv accessed via GBIF.org on 2018-09-21.

Hayashi, K. I. (1999). Crustacea Decapoda: Revision of Pasiphaea sivado (Risso, 1816) and related species, with descriptions of one new genus and five new species (Pasiphaeidae), *In*: Crosnier, A. (Ed.) Résultats des Campagnes MUSORSTOM 20. Mémoires du Muséum national d'Histoire naturelle. Série A, *Zoologie* 180, 267-302.

Holthuis, L. B. & Manning, R. B. (1990). Crabs of the subfamily Dorippinae MacLeay, 1838, from the Indo-West Pacific region (Crustacea: Decapoda: Dorippidae). *Researches on Crustacea* 3, 1-151.

Holthuis, L. B. (2002). The Indo-Pacific scyllarine lobsters (Crustacea, Decapoda, Scyllaridae). *Zoosystema* 24, 499-683.

idigbio.org/portal (2018),
Query: {"filtered": {"filter": {"and": [{"term": {"country": "mozambique"}}, {"query": {"match": {"_all": {"operator": "and", "query": "decapoda"}}}}, {"term": {"order": "decapoda"}}]}}},
349 records, accessed on 2018-11-07T07:42:11.718280, contributed by 11 Recordsets, Recordset identifiers:
http://www.idigbio.org/portal/recordsets/a6eee223-cf3b-4079-8bb2-b77dad8cae9d (147 records)
http://www.idigbio.org/portal/recordsets/3c9420c9-c4a8-47dc-88b7-b5638ca5e716 (109 records)
http://www.idigbio.org/portal/recordsets/b688c17c-3761-4ccd-a42f-88219f5fcff4 (61 records)
http://www.idigbio.org/portal/recordsets/e1e0f2cc-a50c-40d4-9031-c2d90a826247 (15 records)
http://www.idigbio.org/portal/recordsets/271a9ce9-c6d3-4b63-a722-cb0adc48863f (7 records)
http://www.idigbio.org/portal/recordsets/14312086-24f8-453a-be6f-d7a0c796a116 (2 records)

http://www.idigbio.org/portal/recordsets/56879e73-bf9d-4bd9-a7a4-4f2f940d0f62
(2 records)
http://www.idigbio.org/portal/recordsets/5aef068f-efd3-4851-a623-f542e97350cd
(2 records)
http://www.idigbio.org/portal/recordsets/abbf4722-c63b-492f-b183-cb45ad9f5211
(2 records)
http://www.idigbio.org/portal/recordsets/24aa66d3-86a3-44a7-b6a3-6a45edc1e6a9
(1 records)
http://www.idigbio.org/portal/recordsets/b5f4526b-f4fb-4d90-8ce0-975e0cda8ff6
(1 records)

idigbio.org/portal (2018),
Query: {"filtered": {"filter": {"and": [{"term": {"country": "s\u00e3o tom\u00e9"}}, {"query": {"match": {"_all": {"operator": "and", "query": "decapoda"}}}}, {"term": {"order": "decapoda"}}]}}},
1 records, accessed on 2018-11-07T07:49:44.854406, contributed by 1 Recordsets, Recordset identifiers:
http://www.idigbio.org/portal/recordsets/b688c17c-3761-4ccd-a42f-88219f5fcff4
(1 records)

IFREMER - French Research Institute for Exploitation of the Sea. Ifremer BIOCEAN database (Deep Sea Benthic Fauna). Occurrence dataset DOI: 10.15468/yxphxa accessed via GBIF.org on 2019-09-04.

iNaturalist.org (2019). iNaturalist Research-grade Observations. Occurrence dataset DOI: 10.15468/ab3s5x accessed via GBIF.org on 2018-09-21.

iNaturalist.org (2019). iNaturalist Research-grade Observations. Occurrence dataset DOI: 10.15468/ab3s5x accessed via GBIF.org on 2018-09-21.

iNaturalist Research-grade Observations
Observation 3180762, 578653, 10799914, 10859097, 10906644, 10906652, 10906740, 10920406, 11044277, 11064553, 11066625, 11066635, 11066635, 11067095, 11068432, 11147794, 11149022, 11156370, 11290973, 11331585, 11335455, 16230510, 5056450, 6445015, 7293455, 7494751, 7497279, 7498254, 7891236, 9458369, 9561598.

Indian Ocean Biogeographic Information System (IndOBIS) - Distribution records of marine organisms from the Indian Ocean accessed via iOBIS on 2018-11-03.

iziko South African Museum - Crustacean Collection accessed via iOBIS on 2018-11-03.

Klein, E. (2018). NaGISA Project. Caribbean OBIS Node. Occurrence dataset DOI: 10.15468/3gdwj2 accessed via GBIF.org on 2018-09-21.

Komai, T. & Chan, T. Y. (2010). Two new pandalid shrimps and the discovery of the second specimen of the rare hippolytid shrimp Leontocaris bulga Taylor &

Poore, 1998 (Crustacea, Decapoda) from the Mozambique MAINBAZA Cruise. Zoosystema 32, 625-641. DOI: 10.5252/z2010n4a6

Lai, J., Ng, P. & Davie, P. (2010). A revision of the Portunus pelagicus (Linnaeus, 1758) species complex (Crustacea: Brachyura: Portunidae), with the recognition of four species. *THE RAFFLES BULLETIN OF ZOOLOGY* 58.

Lemaitre, R. (2004). A worldwide review of hermit crab species of the genus Sympagurus Smith, 1883 (Crustacea: Decapoda: Parapaguridae). *Mémoires du Muséum National d' Historie Naturelle* 191.

Lemaitre, R, Rahayu, D. L. & Tomoyuki, K. (2018). A revision of "blanket-hermit crabs" of the genus Paguropsis Henderson, 1888, with the description of a new genus and five new species (Crustacea, Anomura, Diogenidae). *ZooKeys* 752, 17-97. DOI: 10.3897/zookeys.752.23712

Litulo, C. & Tudge, C. (2005). Population structure and breeding season of the hermit crab Diogenes brevirostris Stimpson, 1858 (Decapoda, Anomura, Diogenidae) from southern Mozambique. *Journal of Natural History* 39, 2887-2899. DOI: 10.1080/00222930500206107

Litulo, C. (2006). Population and reproductive biology of the fiddler crab Uca chlorophthalmus (Brachyura: Ocypodidae) from Inhaca Island, southern Mozambique. *Journal of the Marine Biological Association of the United Kingdom 86*, 737-742. DOI: 10.1017/S0025315406013646

Litulo, C., Abreu, D. & Tudge, C. C. (2006). Reproductive biology of Calcinus laevimanus (Randall, 1840) (Decapoda: Anomura: Diogenidae) from Inhaca Island, southern Mozambique. Volume Special 2006, 6, 57-66. DOI: 10.18353/crustacea.Special2006.6_57

Litulo, C., Macia, A. & Mantelatto, F. (2010). Fecundity and sexual maturity of the crab Macrophthalmus depressus (Brachyura: Ocypodidae) from Inhaca Island, Mozambique. *African Journal of Aquatic Science* August 2005, 179-183. DOI: 10.2989/16085910509503854.

Macia, A., Quincardete, I. & Paula, J. (2001). A comparison of alternative methods for estimating population density of the fiddler crab Uca annulipes at Saco Mangrove, Inhaca Island (Mozambique). *Hydrobiologia* 449, 213-219. DOI: 10.1023/A:1017502817087

Macpherson, E., Flores, P. R. & Machordom, A. (2017). New sibling species and new occurrences of squat lobsters (Crustacea, Decapoda) from the western Indian Ocean. *European Journal of Taxonomy*. DOI: 10.5852/ejt.2017.343

Maia, H. A., Morais, R. A., Quimbayo, J. P., *et al.* (2019). Spatial patterns and drivers of fish and benthic reef communities at São Tomé Island, Tropical Eastern Atlantic. *Marine Ecology 2018* 39, e12520. DOI: 10.1111/maec.12520

Marine and Coastal Management - Linefish Dataset accessed via iOBIS on 2018-11-03.

Massart, O. (2011). Opportunistic marine mammal sightings near Pemba, Mozambique. Data downloaded from OBIS-SEAMAP (http://seamap.env.duke.edu/dataset/784) on 2018-11-03.

Mkare, T., Groeneveld, J., Teske, P. & Matthee, C. (2017). Comparative genetic structure in two high-dispersal prawn species from the south-west Indian Ocean. *African Journal of Marine Science* 39. DOI: 10.2989/1814232X.2017.1402089.

MNHN - Museum national d'Histoire naturelle (2019). The crustaceans collection (IU) of the Muséum national d'Histoire naturelle (MNHN - Paris). Version 68.131. Occurrence dataset DOI: 10.15468/qgvvhd accessed via GBIF.org on 2018-09-21.

Murina, G.V. (2013). Zoobenthos data from different sources (collected and extracted from literature), personal archive of G.V. Murina (IBSS). Dataset published in electronic format by IBSS in 2013 accessed via iOBIS on 2018-11-03.

Museum and Art Gallery of the Northern Territory (2018). Northern Territory Museum and Art Gallery provider for OZCAM. Occurrence dataset DOI: 10.15468/giro3a accessed via GBIF.org on 2018-09-21.

Muséum national d'Histoire naturelle, Paris (France), Collection: Crustaceans (IU) Specimen MNHN-IU-2008-10100, 10101, 10102, 10103, 10104, 10105, 10106, 10107, 10109, 10110, 10111, 10112, 10113, 10114, 10115, 10116, 10117, 10118, 10119, 10120, 10120, 10121, 10122, 10123, 10124, 10125, 10126, 10128, 10129, 10130, 10131, 10132, 10133, 10134, 10135, 10136, 10139, 10140, 10141, 10142, 10143, 10144, 10145, 10146, 10147, 10148, 10149, 10151, 10153, 10154, 10155, 10157, 10158, 10160, 10161, 10162, 10163, 10164, 10165, 10166, 10167, 10168, 10169, 10170, 10171, 10172, 10173, 10174, 10175, 10176, 10177, 10178, 10179, 10180, 10181, 10182, 10183, 10184, 10185, 10186, 10187, 10188, 10189, 10190, 10191, 10192, 10193, 10194, 10195, 10196, 10197, 10198, 10199, 10200, 10201,10202, 10203, 10204, 10205, 10206, 10207, 10208, 10210, 10212, 10213, 10214, 10215, 10216, 10217, 10218, 10219, 10220, 10221, 10222, 10223, 10224, 10225, 10226, 10227, 10228, 10229, 10230, 10231, 10232, 10233, 10234, 10235, 10237, 10238, 10239, 10241, 10243, 10244, 10246, 10247, 10248, 10249, 10250, 10251, 10252, 10253, 10254, 10255, 10256, 10257, 10258, 10259, 10260, 10261, 10262, 10263, 10264, 10265, 10266, 10267, 10268, 10269, 10270, 10271, 10272, 10273, 10274, 10275, 10276, 10277, 10278, 10279, 10280, 10281, 10282, 10283, 10284, 10285, 10286, 10287, 10288, 10289, 10290, 10291, 10292, 10293, 10294, 10295, 10296, 10297, 10298, 10299, 10300, 10301, 10302, 10303, 10304, 10305, 10306, 10307, 10308, 10309, 10310, 10311, 10312, 10313, 10314, 10315, 10316, 10317, 10318, 10319, 10320, 10321, 10322, 10323, 10324, 10325, 10326, 10327, 10328,

10329, 10330, 10331, 10334, 10335, 10336, 10337, 10338, 10339, 10340, 10341, 10342, 10343, 10344, 10345, 10346, 10347, 10348, 10349, 10350, 10351, 10353, 10354, 10355, 10356, 10357, 10358, 10359, 10361, 10362, 10363, 10364, 10365, 10366, 10367, 10368, 10369, 10370, 10371, 10372, 10373, 10374, 10375, 10376, 10377, 10379, 10381, 10382, 10383, 10386, 10387, 10435, 10436, 10437, 10438, 10439, 10440, 10441, 10442, 10664, 12661, 12662, 12663, 12664, 12665, 12666, 12667, 12668, 12670, 12672, 12673, 12674, 12675, 12677, 12678, 12679, 12680, 12682, 12683, 13172, 13173, 14555, 14556, 14557, 14558, 14559, 14560, 14562, 14563, 14564, 14565, 14566, 14567, 14568, 14569, 14570, 14571, 14572, 14574, 14575, 14576, 14577, 14578, 14579, 14580, 14581, 14582, 14583, 14586, 14587, 14588, 14589, 14590, 14591, 14592, 14593, 14594, 14595, 14596, 14597, 14598, 14599, 14600, 14600, 14601, 14602, 14603, 14604, 14605, 14606, 14607, 14608, 14609, 14610, 14611, 14613, 14615, 14616, 14620, 14622, 14623, 14624, 14625, 20703, 20704, 20705;
Specimen MNHN-IU-2009-1998, 1999, 3959, 3960, 3961, 5744, 5745, 5746, 5747, 5750;
Specimen MNHN-IU-2010-145, 1500, 1501, 1502, 1503, 1504, 1507, 1508, 1509, 1510, 1511, 1512, 1513, 1514, 1515, 1516, 1517, 1518, 1519, 1520, 1521, 1522, 1523, 1524, 1525, 1526, 1527, 1528, 1529, 1530, 1531, 1532, 1533, 1534, 1535, 1536, 1537, 1538, 1539, 1540, 1541, 1542, 1543, 1544, 1545, 1546, 1547, 1548, 1549, 1550, 1551, 1552, 1553, 1554, 1555, 1556, 1557, 1558, 1559, 156, 1560, 1561, 1562, 1563, 157, 158, 159, 160, 161, 162, 164, 177, 186, 187, 1908, 191, 192, 193, 195, 300, 301, 302, 303, 4103, 4104, 4105, 4106, 4150, 4183, 4321, 4322, 4323, 4324, 4326, 4327, 5444, 5445, 5446, 5447, 5449, 5450, 5451, 5452, 5453, 5454, 6133, 6135, 6136, 6137, 7922, 7923, 7924, 8064, 8065, 8066,
Specimen MNHN-IU-2011-5332, 5333, 5334, 5335, 5568, 5569, 5570, 5571, 5989, 5999, 6000, 6001;
Specimen MNHN-IU-2012-142;
Specimen MNHN-IU-2013-11319, 12042, 13217, 13301, 13975, 13976,
14055, 14350, 14351, 14352, 14353, 14354, 14355, 14356, 14357, 14360, 14371, 14372, 14373, 14374, 14375, 14376, 14377, 14388, 14679, 14680, 16002,16003, 16004, 16005, 16006, 16058, 16059, 16060, 16677, 16738, 16739, 16744, 19772, 19773, 19774, 19775, 19776, 19777, 19778, 19779, 19780, 19781, 19782, 19783, 19784, 19785, 19786, 19787, 19788, 19789, 19790, 19791, 19792, 19793, 19794, 19795, 5998, 5999, 6937, 6938, 6939, 6946, 6947, 6948, 6949, 6950, 6951, 6952, 6953, 6954, 6955, 6956, 7186, 7362, 7364, 7366, 7373, 7374, 7376, 7377, 8306, 8308, 8309, 8404, 8405, 8407, 9649, 9650, 9651, 9652, 9653, 9654, 9655, 9656, 9657, 9658, 9659, 9660, 9669, 9758, 9976, 9977;
Specimen MNHN-IU-2014-10109, 10252, 10253, 10254, 10255, 10337,
10338, 10339, 10340, 10341, 10342, 10561, 11094, 13477, 13478, 13479, 13503, 13504, 13505, 13522, 13523, 13525, 13526, 13544, 13546, 13549, 13550, 13551, 13552, 13555, 13556, 13571, 13573, 13573, 13574, 13576, 13578, 13598, 13599, 13611, 13612, 17466, 19214, 19215, 19216, 19257, 19258, 20568, 20569, 22263, 22264, 22283, 22802, 22803, 22805, 22807,

22808, 22809, 22810, 22811, 22812, 22813, 5092, 5686, 6247, 6254, 6311, 6312, 6313, 6314, 6315;
Specimen MNHN-IU-2016-11806, 11809, 11810, 11811, 11812, 1335, 161, 2944, 2946, 2949, 3569, 3571, 3620, 3730, 3739, 3764, 7230, 7231, 8857, 9286, 9293, 9294, 9295, 9296, 9297, 9298, 9299, 9300, 9301, 9302, 9303, 9304, 9308, 9309, 9310;
Specimen MNHN-IU-2017-8824.

National Chemical Laboratory. IndOBIS, Indian Ocean Node of OBIS. Occurrence dataset DOI: 10.15468/tbedgi accessed via GBIF.org on 2018-09-21.

Natural History Museum (2019). Natural History Museum (London) Collection Specimens. Occurrence dataset DOI: 10.5519/0002965 accessed via GBIF.org on 2018-09-21.

Naturalis Biodiversity Center, Zoology and Geology catalogues
Specimen RMNH.CRUS.D.17553, 21360, 21379, 21495, 21527, 27239, 27325, 27327, 27328, 27329, 27341, 27343, 27385, 27401, 27421, 27443, 27636, 27806, 28244, 28245, 28246, 28560, 28561, 32320, 32331, 32347, 32350, 32398, 32423, 32424, 32443, 33136, 33271, 33888, 38193, 38198, 39850, 40839, 40886, 41354, 41355, 42892, 42893, 42894, 46717, 46718, 46720, 46721, 46722, 46723, 48676, 51197, 53441, 54566, 54831, 55093, 55110, 55164, 55168, 55205, 55208;
Specimen ZMA.CRUS.D.202666, 202667, 241168, 241979.

Ng, P. K. L. & Castro, P. (2016). Revision of the family Chasmocarcinidae Serène, 1964 (Crustacea, Brachyura, Goneplacoidea). Plazi.org taxonomic treatments database. Checklist dataset DOI: 10.11646/zootaxa.4209.1.1 accessed via GBIF.org on 2018-09-21.

Obura, D. (2012). The Diversity and Biogeography of Western Indian Ocean Reef-Building Corals. Provided by CORDIO, East Africa. *PLoS ONE* 7(9), e45013. DOI: 10.1371/journal.pone.0045013 accessed via iOBIS on 2018-11-03.

Oceanographic Center of Cadiz, Spanish Institute of Oceanography (IEO). Occurrence DOI: 10.15468/anztjy accessed via GBIF.org on 2018-09-21.

Olivas Gonzlez, F. J. (2016). Biological Reference Collections ICM CSIC. Institute of Marine Sciences (ICM-CSIC) DOI: 10.15470/qlqqdx accessed via iOBIS on 2018-11-03.

Olivas González, F. J. (2018). Biological Reference Collections ICM CSIC. Version 1.23. Institute of Marine Sciences (ICM-CSIC). Occurrence dataset DOI: 10.15470/qlqqdx accessed via GBIF.org on 2018-09-21.

Orrell, T. (2019). NMNH Extant Specimen Records. Version 1.21. National Museum of Natural History, Smithsonian Institution. Occurrence dataset DOI: 10.15468/hnhrg3 accessed via GBIF.org on 2018-09-21.

Paulay, G. Brown, W. (2019). UF Invertebrate Zoology. Florida Museum of Natural History. Occurrence dataset DOI: 10.15468/sm6qo6 accessed via GBIF.org on 2018-09-21.

Poore, G. (1994). A phylogeny of the families of Thalassinidea (Crustacea: Decapoda) with keys to families and genera. *Memoirs of the Museum of Victoria* 54, 79-120. DOI: 10.24199/j.mmv.1994.54.03

Raiva, R., Santana, P. (2019). Diversidade e ocorrência de peixes em Inhambane (2009-2017). Version 1.4. National Institute of Fisheries Research (IIP) – Mozambique. Occurrence dataset https://doi.org/10.15468/4fj2tq accessed via GBIF.org on 2018-09-21.

Relatório do Chefe de Missão de Estudos de Pesca de Angola, Primeira Campanha (23 de Maio de 1951 - 23 de Fevereiro de 1952) accessed via iOBIS on 2018-11-03.

Relatório do Chefe de Missão de Estudos de Pesca de Angola, Segunda Campanha (23 de Fevereiro de 1952 - 5 Julho 1953) accessed via iOBIS on 2018-11-03.

Royal Belgian Institute of Natural Sciences (2017). RBINS DaRWIN. Occurrence dataset DOI: 10.15468/qxy4mc accessed via GBIF.org on 2018-09-21.

Saito, T. & Tomoyuki, K. (2008). A review of species of the genera Spongicola de Haan, 1844 and Paraspongicola de Saint Laurent & Cleva, 1981 (Crustacea, Decapoda, Stenopodidea, Spongicolidae). *Zoosystema* 30, 87-147.

Sakai, K. (2004) Dr. R. Plante's collection of the families Callianassidae and Gourretiidae (Decapoda, Thalassinidea) from Madagascar, with the description of two new genera and one new species of the Gourretiidae Sakai, 1999 (new status) and two new species of the Callianassidae Dana, 1852. *Crustaceana* 77, 553-602.

Sakai, K. (2006) Upogebiidae of the world (Decapoda, Thalassinidea). Crustaceana Monographs. Fransen, C.H.J.M. and J. C. von Vaupel Klein (eds.) 6, 186. Leiden.

Schubart, C. & Cuesta, J. (2010). Phylogenetic relationships of the Plagusiidae Dana, 1851 (Brachyura), with description of a new genus and recognition of Percnidae Števcic, 2005, as an independent family. *Crustaceana*. Monographs 11, 279-299.

Senckenberg. Collection Crustacea SMF. Occurrence dataset DOI: 10.15468/mc7ysi accessed via GBIF.org on 2018-09-21.

Senckenberg. Collection Crustacea - ZIM Hamburg. Occurrence dataset DOI: 10.15468/zxrapv accessed via GBIF.org on 2018-09-21.

Senckenberg. Collection Crustacea - ZMB. Occurrence dataset DOI: 10.15468/fwghff accessed via GBIF.org on 2018-09-21.

Serène, R., & Crosnier, A. (1984). Crustacés décapodes brachyoures de l'océan Indien occidental et de la mer Rouge: Xanthoidea: Xanthidae et Trapeziidae. Addendum: Carpiliidae et Menippidae.

Simbine, L., Marques, C., Freitas, P., Gusmão, J., Tavares, C., Samucidine, K., Galetti, Jr., P. (2018). Metapenaeus dobsoni (Miers, 1878), an alien Penaeidae in Mozambican coastal waters: confirmation by mtDNA and morphology analyses. Project: Genetics Assessment in Litopenaeus vannamei and other shrimps and prawns.

Smithsonian National Museum of Natural History, Research & Collections: Invertebrate Zoology Collections

Specimen USNM 216157, 213458, 213120, 1109175, 216163, 135208, 1100676, 165103, 216129, 165094, 165089, 216150, 135131, 216135, 216137, 216130, 216158, 1109176, 165087, 1100675, 138044, 135114, 213463, 135105, 309742, 184282, 165097, 216156, 165091, 213470, 213461, 135187, 264779, 1123845, 216116, 135238, 216148, 165102, 213466, 135239, 216118, 135176, 216124, 216142, 138238, 1132009, 1123847, 165090, 216152, 216112, 216147, 165099, 216122, 216123, 216117, 270343, 213468, 1109195, 181365, 228310, 1109179, 172159, 180391, 216114, 171610, 216145, 1109178, 172332, 1109196, 270432, 252559, 165095, 310814, 138239, 1007737, 135106, 125480, 213453, 1109199, 1109198, 180386, 1123846, 309743, 213994, 138237, 213465, 135250, 135177, 184550, 1202391, 216162, 216120, 216126, 216164, 135172, 228823, 275704, 254129, 1123866, 213475, 233468, 165088, 165101, 165086, 135232, 204623, 216146, 204639, 165098, 135144, 135151, 135233, 135205, 213457, 213460, 135100, 213455, 216160, 216140, 268772, 216139, 1123867, 138054, 1082168, 1123843, 104741, 213462, 1109174, 1109200, 1188871, 137947, 216159, 1004634, 216151, 104739, 216134, 1100677, 216121, 138240, 216141, 216161, 216111, 213467, 205715, 216154, 1109197, 213469, 216133, 216125, 1132007, 275794, 1109180, 216155, 165100, 180394, 216149, 216115, 1004698, 184552, 165093, 216132, 165092, 135102, 165096, 1109177, 213121, 216136, 213477, 309744, 213464, 1123865.

South African Institute for Aquatic Biodiversity - Fish Collection accessed via iOBIS on 2018-11-03.

Southwestern Pacific OBIS (2014). Biological observations from the Dana Expedition Reports. Southwestern Pacific OBIS, National Institute of Water and Atmospheric Research (NIWA), Wellington, New Zealand, 3101 records, Online http://nzobisipt.niwa.co.nz/resource.do?r=danaexpedition released on January 16, 2018 accessed via iOBIS on 2018-11-03.

Southwestern Pacific OBIS (2014). Biological observations from the Discovery Investigations 1925-1935. Southwestern Pacific OBIS, National Institute of Water and Atmospheric Research (NIWA), Wellington, New Zealand, 33337 records, Online http://nzobisipt.niwa.co.nz/resource.do?r=discovery_reports released on January 23, 2015 accessed via iOBIS on 2018-11-03.

Telenius, A., Shah, M. (2019). Invertebrates Collection of the Swedish Museum of Natural History. GBIF-Sweden. Occurrence dataset DOI: 10.15468/eyda6l accessed via GBIF.org on 2018-09-21.

The Danish Biodiversity Information Facility, Galathea II, Danish Deep Sea Expedition 1950-52 DOI: 10.15468/ouseij accessed via iOBIS on 2018-11-03.

The International Barcode of Life Consortium (2016). International Barcode of Life project (iBOL). Occurrence dataset DOI: 10.15468/inygc6 accessed via GBIF.org on 2018-09-21.

Tomoyuki, K. & Wada, K. (2008). A revision of the estuarine crab genus Ilyograpsus Barnard, 1955 (Crustacea: Decapoda: Brachyura: Macrophthalmidae), with descriptions of a new genus and one new species. *Raffles Bulletin of Zoology*. 56.

Torres, P., Penha-Lopes, G., Macia, A. & Paula, J. (2007). Population structure and egg production of the seagrass shrimp Hippolyte kraussiana Stimpson, 1860 (Decapoda: Hippolytidae) at Inhaca Island, Mozambique. *Invertebrate Reproduction & Development* 50, 145-153. DOI: 10.1080/07924259.2007.9652239.

Torres, P., Penha-Lopes, G., Macia, A. & Paula, J. (2007). Population structure, fecundity and embryo loss of the sea grass shrimp Latreutes pymoeus (Decapoda: Hippolytidae) at Inhaca Island, Mozambique. *Journal of the Marine Biological Association of the United Kingdom* 87, 879-884. DOI: 10.1017/S0025315407056767.

Torres, P. Penha-Lopes, G. & Narciso, L. & Macia, A. & Paula, Jose. (2009). Fecundity and brood loss in four species of fiddler crabs, genus Uca (Brachyura: Ocypodidae), in the mangroves of Inhaca Island, Mozambique. *Journal of The Marine Biological Association of The United Kingdom* 89. DOI: 10.1017/S0025315408002701.

Tuya, F., Herrero-Barrencua, A., Bosch, N. E., Abreu, A. D. & Haroun, R. (2017). Reef fish at a remote tropical island (Principe Island, Gulf of Guinea): disentangling taxonomic, functional and phylogenetic diversity patterns with depth. Marine and Freshwater Research 69, 395-402. DOI: 10.1071/MF17233.

Van Guelpen, L. (2016). Atlantic Reference Centre Museum of Canadian Atlantic Organisms - Invertebrates and Fishes Data. Version 4 In OBIS Canada Digital Collections. Bedford Institute of Oceanography, Dartmouth, NS, Canada accessed via iOBIS on 2018-11-03.

Vassily, A. S. & Michael, A. (2007). A new species and new records of deep-water Calappidae (Crustacea: Decapoda) from the Indian Ocean with a key to the Mursia Desmarest, 1823 species of the region. *Journal of Natural History* 41, 45-48; 2851-2890. DOI: 10.1080/00222930701770786.

Williams, A. B. (1982) Revision of the genus Latreillia Roux (Brachyura: Homoloidea). *Quaderni del Laboratorio di Tecnologia della Pesca* 3(2–5), 227-255.

Wirtz, P. (2008). The Gulf of Guinea goby-shrimp symbiosis and a review of goby thalassinidean associations. Arquipélago: *Life and Marine Sciences* 25.

WoRMS Editorial Board (2017). Type locality distributions from the World Register of Marine Species. Available from http://www.marinespecies.org at VLIZ, accessed via iOBIS on 2018-11-03.

6.2. Code Scripts

6.2.1. getData.py

```
1.  import requests
2.  import csv
3.
4.  """Write specific data from the MNHN collection website.
5.
6.      Requires:
7.      - the input Order and Country to search for.
8.
9.      Ensures:
10.     - a csv file with data about name, locality, event date, depth and coordinates o
    f collection entries.
11.     """
12.
13. Order = input("Insert ORDER: ")
14. Country = input("Insert COUNTRY: ")
15.
16. search = "https://science.mnhn.fr/institution/mnhn/collection/iu/item/list?order="+O
    rder.title()+"&country="+Country.upper()        # website search link
17.
18. links = []                                                  # create list for websit
    e links
19. dataSet = []                                                # create list for data i
    nformation
20. temp = "Name","Locality","Date","Depth","Coordinates"       # create csv file header

21.
22. r = requests.get(search)                                    # retrieve data via web
    request
23. t = r.text                                                  # access the data and tr
    ansform it in raw text
24.
25. myfile = t.split("\n")                                      # split the search terms
     one per line
26.
27. line = 0
28.
29. for i in myfile:                                            # loop to create link to
     access collection entries
30.
31.     if "doc-index" in i:
32.         path = myfile[line+10].split('href="')[1][:-2]
33.         url = "https://science.mnhn.fr" + path
34.         links.append(url)
35.     line = line + 1
36.
37. for i in links:                                             # loop to seek each link

38.
39.     r = requests.get(i)
40.     t = r.text
41.     myfile = t.split("\n")
42.     line = 0
43.
44.     dataSet.append(temp)
45.     temp = []
46.
47.     for j in myfile:                                        # loop to save for each
    entry the intended information
48.         if "<title>" in j:
```

```
49.             name = j[j.find("<title>")+1:j.find("</title>")]
50.             name = name[name.find(";")+1:len(name)]
51.             name = name[name.find("- ")+1:len(name)]
52.             temp.append(name)
53.
54.         if '<div id="locality">' in j:
55.             locality = j[j.find('<div id="locality">')+19:j.find("</div>")].rstrip("
    \n")
56.             temp.append(locality)
57.
58.         if 'Verbatim locality' in j:
59.             temp.pop()
60.
61.         if '<div id="eventDate">' in j:
62.             date = myfile[line+1][:-6].lstrip()
63.             tdate = date
64.             if len(temp) == 2:
65.                 temp.append(date)
66.             else:
67.                 temp.append("")
68.                 temp.append(date)
69.
70.         if '<div id="depth">' in j:
71.             depth = myfile[line+1].lstrip() + myfile[line+2][:-6].strip()
72.             if "div" in depth:
73.                 depth = depth[:-6].strip()
74.
75.             if len(temp) == 3:
76.                 temp.append(depth)
77.             else:
78.                 while len(temp) < 3:
79.                     temp.append("")
80.                 temp.append(depth)
81.
82.         if '<div id="decimalCoordinates" style="visibility: hidden;">' in j:
83.             coord = myfile[line+1].lstrip() + myfile[line+2][:-6].strip()
84.             tcoord = coord
85.             if len(temp) == 4:
86.                 temp.append(coord)
87.             else:
88.                 while len(temp) < 4:
89.                     temp.append("")
90.                 temp.append(coord)
91.
92.         line += 1
93.
94. dataSet.append(temp)                                    # append the information retriev
    ed into variable tempo into dataSet
95.
96. with open('DataSet.csv', 'w') as csvFile:
97.     writer = csv.writer(csvFile)
98.     writer.writerows(dataSet)
99.
100.        csvFile.close()                                      # write in csv file
101.
102.        print("File Write Completed!")
```

6.2.2. previousName.py

```
1.  import csv
2. from suds import null, WebFault
3. from suds.client import Client
4. from socket import timeout
5.
6. """Check in WoRMS website if a scientific name is accepted or unaccepted.
7.
8.     Requires:
9.     - the dataset csv file.
10.
11.    Ensures:
12.    - a csv file with two columns: the accepted scienfitic name and the previous una
   ccepted scientific name.
13.    """
14.
15. pathFile = input("Insert Dataset File Name or File Directory: ")
16.
17.
18. cl = Client('http://www.marinespecies.org/aphia.php?p=soap&wsdl=1')         # WoRMS
   webservice to feed with standard WoRMS data
19.
20. cl.set_options(timeout=1000)
21.
22. Output = []                                                                 # create
    list for file output
23.
24. line_count = 0                                                              # line c
   ount until it stops running
25.
26. Prev_Run = 0                                                                # line i
   ndex of the previous run
27.
28. if Prev_Run == 0:                                                           # check
   previously run
29.     Output.append(("scientificName", "previousName"))                       # output
    file headers
30.
31. with open(pathFile) as csv_file:
32.     csv_reader = csv.reader(csv_file, delimiter=';')
33.
34.     for row in csv_reader:                                                  # loop t
   o seek the input file
35.
36.         line_count+=1
37.
38.         if line_count > Prev_Run:                                           # calli
   ng the matchAphiaRecordsByNames function from Python
39.
40.             print('linha: %s' %(line_count-1))
41.
42.             st = 1
43.
44.             scinames = cl.factory.create('scientificnames')
45.             scinames["_arrayType"] = "string[]"
46.             scinames["scientificname"] = row[1]
47.
48.             try:
49.                 array_of_results_array = cl.service.matchAphiaRecordsByNames(sciname
   s, like="true", fuzzy="false", marine_only="true")
50.             except:
51.                 with open('Result.csv', 'a+', encoding='utf-8') as csvFile: #
52.                     writer = csv.writer(csvFile, delimiter=';')
53.                     writer.writerows(Output)
```

```
54.                     csvFile.close()
55.                     quit()
56.
57.             for results_array in array_of_results_array:                    # loop t
    o seek arrays in array for each entry
58.                 for aphia_object in results_array:
59.                     if aphia_object.status != "accepted":
60.                             Prev_Name = (aphia_object.valid_name + " " + aphia_objec
    t.valid_authority)
61.                     else:
62.                         Prev_Name = ''
63.                     if st == 1:
64.                         Output.append((row[1], Prev_Name))                  # append
     the information retrieved into Output
65.                         st=0
66.
67. csv_file.close()
68.
69. with open('previousName.csv', 'a+', encoding='utf-
    8') as csvFile:           # write csv file
70.     writer = csv.writer(csvFile, delimiter=';')
71.     writer.writerows(Output)
72.
73. csvFile.close()
```

6.2.3. taxonInfo.py

```
import csv
from suds import null, WebFault
from suds.client import Client
from socket import timeout

"""Get scientific name URL links from WoRMS webiste.

    Requires:
    - the dataset csv file with the correct header.

    Ensures:
    - a csv file with the same structure as the dataset with the taxonInfo column filled.
    """

pathFile = input("Insert Dataset File Name or File Directory: ")

cl = Client('http://www.marinespecies.org/aphia.php?p=soap&wsdl=1')             # WoRMS webservice to feed with standard WoRMS data

cl.set_options(timeout=1000)


Output = []                                                                    # create list for file output

line_count = 0                                                                 # line count until it stops running

Prev_Run = 0                                                                   # line index of the previous run

if Prev_Run == 0:                                                              # check previously run
    Output.append(("#CatalogNumber", "scientificName", "kingdom", "phylum", "class", "order", "family", "genus", "species", "infraspecificepithet", "taxonRank", "previousName",
        "country", "locality", "verbatimLocality", "habitat", "decimalLatitude", "decimalLongitude", "geodeticDatum", "depth", "individualCount", "eventDate", "institutionCode",
        "occurrenceID", "references", "taxonInfo"))                            # output file headers

with open(pathFile) as csv_file:
    csv_reader = csv.reader(csv_file, delimiter=';')

    for row in csv_reader:                                                     # loop to seek the input file

        line_count+=1

        if line_count > Prev_Run:                                               # calling the matchAphiaRecordsByNames function from Python

            print('linha: %s' %line_count)

            st = 1

            scinames = cl.factory.create('scientificnames')
            scinames["_arrayType"] = "string[]"
            scinames["scientificname"] = row[1]

            try:
```

```
                array_of_results_array = cl.service.matchAphiaRecordsByNames(sciname
s, like="true", fuzzy="false", marine_only="false")
            except:
                with open('Result.csv', 'a+', encoding='utf-8') as csvFile:
                    writer = csv.writer(csvFile, delimiter=';')
                    writer.writerows(Output)
                    csvFile.close()
                    quit()

            for results_array in array_of_results_array:                        # lo
op to seek arrays in array for each entry
                for aphia_object in results_array:
                    if st == 1:
                        Output.append((row[0], row[1], row[2], row[3], row[4], row[5
], row[6], row[7], row[8], row[9], row[10], row[11], row[12], row[13], row[14],
                            row[15], row[16], row[17], row[18], row[19], row[20], ro
w[21], row[22], row[23], row[24], "http://www.marinespecies.org/aphia.php?p=taxdetai
ls&id="+str(aphia_object.AphiaID)))
                        st=0                                                    # ap
pend the information retrieved into Output

csv_file.close()

with open('Result.csv', 'a+', encoding='utf-
8') as csvFile:                     # write csv file
    writer = csv.writer(csvFile, delimiter=';')
    writer.writerows(Output)

csvFile.close()
```

6.2.4. coordDec.py

```
1.  import sys
2.
3.  """Convert Degrees Minutes Seconds (DMS) coordinates to Decimal.
4.
5.      Requires:
6.      - a txt file with DMS coordinates per line formatted e.g. as following: 7° 20' 0
    '' E
7.
8.      Ensures:
9.      - a txt file with Decimal coordinates.
10.     """
11.
12. if len(sys.argv) < 2:                                  # display message "File not fou
    nd!" when missing the input file
13.     print("File not found!")
14.     quit()
15.
16. in_file = open(sys.argv[1], 'r')                       # input file given with program
    call
17.
18. out_file=open("Decimal.txt", 'w')                      # output file named Decimal.txt

19.
20. for line in in_file:                                   # loop to split DMS coordinates
    in variables degrees (D), minutes (M) and seconds (S)
21.                                                        # to apply the following express
    ion: degrees + (minutes/60) + (seconds/3600)
22.     c = line.split()
23.
24.     D = float(c[0][:-1])
25.
26.     M = 0
27.     if len(c) > 2:
28.         M = float(c[1][:-1])/60
29.
30.     S = 0
31.     if len(c) > 3:
32.         S = float(c[2][:-2])/3600
33.
34.     Dec = D + M + S
35.
36.     if c[len(c)-1] == 'S' or c[len(c)-
    1] == 'W':    # verify if the coordinate is south or west, if it's either transforms
     the coordinate in negative value
37.         Dec = 0 - Dec
38.
39.     out_file.write("%.4f" % Dec+"\n")
40.
41. in_file.close()
42.
43. out_file.close()
```

6.2.5. matchShape.py

```
1.  import csv
2.  import shapefile
3.  from shapely.geometry import Point # Point class
4.  from shapely.geometry import shape # shape() is a function to convert geo objects th
    rough the interface
5.
6.  """Add habitat type to point coordinates.
7.
8.      Requires:
9.      - the dataset csv file with the correct header.
10.     - habitats shapefiles: coral, mangrove and seagrass
11.     - countries shapefile.
12.
13.     Ensures:
14.     - a csv file with the same structure as the dataset with the habitat column fill
    ed.
15.     """
16.
17. # one input for each required file
18. pathFileIL = input("Insert shapefile Inland File Name or File Directory: ")
19.
20. pathFileMG = input("Insert shapefile Mangrove File Name or File Directory: ")
21.
22. pathFileMGpt = input("Insert shapefile Mangrove Points File Name or File Directory:
    ")
23.
24. pathFileSG = input("Insert shapefile Seagrass Layer File Name or File Directory: ")

25.
26. pathFileSGpt = input("Insert shapefile Seagrass Points File Name or File Directory:
    ")
27.
28. pathFileCR = input("Insert shapefile Coral File Name or File Directory: ")
29.
30. pathFileCRpt = input("Insert shapefile Coral Points File Name or File Directory: ")

31.
32. pathFileDS = input("Insert Dataset File Name or File Directory: ")
33.
34.
35. shp = shapefile.Reader(pathFileIL) #open the shapefile
36. Inland = shp.shapes() # get all the polygons
37.
38. shp = shapefile.Reader(pathFileMG) #open the shapefile
39. MG = shp.shapes() # get all the polygons
40.
41. shp = shapefile.Reader(pathFileMG) #open the shapefile
42. MGpt = shp.shapes() # get all the points
43.
44. shp = shapefile.Reader(pathFileSG) #open the shapefile
45. SG = shp.shapes() # get all the polygons
46.
47. shp = shapefile.Reader(pathFileSGpt) #open the shapefile
48. SGpt = shp.shapes() # get all the points
49.
50. shp = shapefile.Reader(pathFileCR) #open the shapefile
51. CR = shp.shapes() # get all the polygons
52.
53. shp = shapefile.Reader(pathFileCR) #open the shapefile
54. CRpt = shp.shapes() # get all the points
55.
56. Output = []
57.
```

```
s=0

# output file headers
Output.append(("#catalogNumber", "scientificName", "kingdom", "phylum", "class", "order", "family", "genus", "species", "infraspecificepithet", "taxonRank", "previousName",
    "countryCode", "locality", "verbatimLocality" "habitat", "decimalLatitude", "decimalLongitude", "geodeticDatum", "depth", "individualCount", "eventDate", "institutionCode",
    "occurrenceID", "references"))

with open(pathFileDS) as csv_file:
    csv_reader = csv.reader(csv_file, delimiter=';')
    line_count = 0

    for row in csv_reader:
        if line_count == 0:
            line_count += 1
        else:
            pt = Point(float(row[17]), float(row[16]))
            print("Ponto #", (row[0], row[17], row[16]))
            In = 0

            if In == 0:
                print("Layer: Coral")
                for i in (CR):
                    if (shape(i).distance(pt)) < 0.005: # checks whether the coordinates points from the dataset are at chosen distance (=0.005) from coral polygons and appends the result to the habitat column
                        Output.append((row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[11], row[12], row[13], "Coral",
                            row[15], row[16], row[17], row[18], row[19], row[20], row[21], row[22], row[23], row[24]))
                        In = 1
                        break

            if In == 0:
                print("Points: Coral")
                for s in (CRpt):
                    if (shape(s).distance(pt)) < 0.005: # checks whether the coordinates points from the dataset are at chosen distance (=0.005) from coral points and appends the result to the habitat column
                        Output.append((row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[11], row[12], row[13], "Seagrass",
                            row[15], row[16], row[17], row[18], row[19], row[20], row[21], row[22], row[23], row[24]))
                        In = 1
                        break

            if In == 0:
                print("Layer: Mangrove")
                for k in (MG):
                    if (shape(k).distance(pt)) < 0.005: # checks whether the coordinates points from the dataset are at chosen distance (=0.005) from mangrove polygons and appends the result to the habitat column
                        Output.append((row[0], row[1], row[2], row[3], row[4], row[5], row[6], row[7], row[8], row[9], row[10], row[11], row[12], row[13], "Mangrove",
                            row[15], row[16], row[17], row[18], row[19], row[20], row[21], row[22], row[23], row[24]))
                        In = 1
                        break

                if In == 0:
                    print("Points: Mangrove")
```

```
106.                         for s in (MGpt):
107.                             if (shape(s).distance(pt)) < 0.005: # checks whether the
    coordinates points from the dataset are at chosen distance (=0.005) from mangrove po
    ints and appends the result to the habitat column
108.                                 Output.append((row[0], row[1], row[2], row[3], row[4]
    , row[5], row[6], row[7], row[8], row[9], row[10], row[11], row[12], row[13], "Seagr
    ass",
109.                                     row[15], row[16], row[17], row[18], row[19], row[
    20], row[21], row[22], row[23], row[24]))
110.                                 In = 1
111.                                 break
112.
113.                     if In == 0:
114.                         print("Layer: Seagrass")
115.                         for j in (SG):
116.                             if (shape(j).distance(pt)) < 0.005: # checks whether the
    coordinates points from the dataset are at chosen distance (=0.005) from seagrass po
    lygons and appends the result to the habitat column
117.                                 Output.append((row[0], row[1], row[2], row[3], row[4]
    , row[5], row[6], row[7], row[8], row[9], row[10], row[11], row[12], row[13], "Seagr
    ass",
118.                                     row[15], row[16], row[17], row[18], row[19], row[
    20], row[21], row[22], row[23], row[24]))
119.                                 In = 1
120.                                 break
121.
122.                     if In == 0:
123.                         print("Points: Seagrass")
124.                         for s in (SGpt):
125.                             if (shape(s).distance(pt)) < 0.005: # checks whether the
    coordinates points from the dataset are at chosen distance (=0.005) from seagrass po
    ints and appends the result to the habitat column
126.                                 Output.append((row[0], row[1], row[2], row[3], row[4]
    , row[5], row[6], row[7], row[8], row[9], row[10], row[11], row[12], row[13], "Seagr
    ass",
127.                                     row[15], row[16], row[17], row[18], row[19], row[
    20], row[21], row[22], row[23], row[24]))
128.                                 In = 1
129.                                 break
130.
131.                     if In ==0:
132.                         print("Layer: Costal Area")
133.                         for l in (Inland):
134.                             if (pt).within(shape(l)): # checks whether the coordinate
    s points from the dataset are at chosen distance (=0.005) from seagrass points and a
    ppends the result to the habitat column
135.                                 Output.append((row[0], row[1], row[2], row[3], row[4]
    , row[5], row[6], row[7], row[8], row[9], row[10], row[11], row[12], row[13], "Costa
    l Area",
136.                                     row[15], row[16], row[17], row[18], row[19], row[
    20], row[21], row[22], row[23], row[24]))
137.                                 In = 1
138.                                 break
139.
140.                     if In == 0:
141.                         print("Open Sea") # appends the result to the habitat column

142.                         Output.append((row[0], row[1], row[2], row[3], row[4], row[5]
    , row[6], row[7], row[8], row[9], row[10], row[11], row[12], row[13], "Open Sea",
143.                             row[15], row[16], row[17], row[18], row[19], row[20], row
    [21], row[22], row[23], row[24]))
144.
145.       csv_file.close()
146.
147.       with open('Habitats.csv', 'w') as csvFile:        # write results in CSV file
```

```
148.            writer = csv.writer(csvFile, delimiter=';')
149.            writer.writerows(Output)
150.
151.        csvFile.close()
```

6.2.6. PostgreSQL

```
Create TABLE occurrences(
catalogNumber integer,
depthinMeters float,
individualCount integer not null,
eventDate date,
PRIMARY KEY (catalogNumber)
);

Create TABLE species(
catalogNumber integer,
scientificName text not null,
kingdom text not null,
phylum text not null,
class text not null,
“order” text not null,
family text not null,
genus text not null,
species text,
infraspecificepithet text,
taxonRank text not  null,
previousName text,
taxonInfo text not null,
“type” text not null,
PRIMARY KEY (catalogNumber),
CONSTRAINT fk_sp FOREIGN KEY (catalogNumber) REFERENCES occurrences(catalogNumber)
);

Create TABLE location(
catalogNumber integer,
country text not null,
locality text not null,
verbatimLocality text,
habitat text not null,
decimalLatitude float not null,
decimalLongitude float not null,
geodeticDatum text not null,
PRIMARY KEY (catalogNumber),
CONSTRAINT fk_lo FOREIGN KEY (catalogNumber) REFERENCES occurrences(catalogNumber)
);

Create TABLE source(
catalogNumber integer,
institutionCode text,
occurrenceID text,
“references” text,
PRIMARY KEY (catalogNumber),
CONSTRAINT fk_so FOREIGN KEY (catalogNumber) REFERENCES occurrences(catalogNumber)
);


Alter table location add column coordinates geometry (Point, 4326);
UPDATE location SET coordinates = ST_SetSRID(ST_MakePoint(
decimallongitude, decimallatitude), 4326);


CREATE VIEW view_search AS
SELECT COALESCE(scientificname) || COALESCE(' (#') || COALESCE(catalognumber) || COA
LESCE(')') as search,
species.catalognumber
from species;
```

6.2.7. Leaflet

```
<!doctype html>
<html lang="en">
    <head>
        <meta charset="utf-8">
        <meta http-equiv="X-UA-Compatible" content="IE=edge">
        <meta name="viewport" content="initial-scale=1,user-scalable=no,maximum-scale=1,width=device-width">
        <meta name="mobile-web-app-capable" content="yes">
        <meta name="apple-mobile-web-app-capable" content="yes">
        <link rel="stylesheet" href="css/leaflet.css">
        <link rel="stylesheet" href="css/qgis2web.css"><link rel="stylesheet" href="css/fontawesome-all.min.css">
        <link rel="stylesheet" href="css/MarkerCluster.css">
        <link rel="stylesheet" href="css/MarkerCluster.Default.css">
        <link rel="stylesheet" href="css/leaflet-search.css">
        <link rel="stylesheet" href="css/leaflet-control-geocoder.Geocoder.css">
        <style>
        html, body, #map {
            width: 100%;
            height: 100%;
            padding: 0;
            margin: 0;
        }
        </style>
        <title></title>
    </head>
    <body>
        <div id="map">
        </div>
        <script src="js/qgis2web_expressions.js"></script>
        <script src="js/leaflet.js"></script>
        <script src="js/multi-style-layer.js"></script>
        <script src="js/leaflet-svg-shape-markers.min.js"></script>
        <script src="js/leaflet.rotatedMarker.js"></script>
        <script src="js/leaflet.pattern.js"></script>
        <script src="js/leaflet-hash.js"></script>
        <script src="js/Autolinker.min.js"></script>
        <script src="js/rbush.min.js"></script>
        <script src="js/labelgun.min.js"></script>
        <script src="js/labels.js"></script>
        <script src="js/leaflet-control-geocoder.Geocoder.js"></script>
        <script src="js/leaflet.markercluster.js"></script>
        <script src="js/leaflet-search.js"></script>
        <script src="data/Coastline_0.js"></script>
        <script src="data/STPEEZ_1.js"></script>
        <script src="data/STPoutline_2.js"></script>
        <script src="data/STParea_3.js"></script>
        <script src="data/STPboundaries_4.js"></script>
        <script src="data/MOZEEZ_5.js"></script>
        <script src="data/MOZoutline_6.js"></script>
        <script src="data/MOZarea_7.js"></script>
        <script src="data/MOZboundaries_8.js"></script>
        <script src="data/Seagrasspointdata_9.js"></script>
        <script src="data/Seagrasspolygondata_10.js"></script>
        <script src="data/Mangrovepointdata_11.js"></script>
        <script src="data/Mangrovepolygondata_12.js"></script>
        <script src="data/Coralpointdata_13.js"></script>
        <script src="data/Coralpolygondata_14.js"></script>
        <script src="data/Speciesoccurrences_15.js"></script>
        <script>
        var map = L.map('map', {
            zoomControl:true, maxZoom:28, minZoom:1
```

```
        }).fitBounds([[-38.35598548426972,-18.600019343159722],[21.739020714643903,73.15140111972579]]);
        var hash = new L.Hash(map);
        map.attributionControl.setPrefix('<a href="https://github.com/tomchadwin/qgis2web" target="_blank">qgis2web</a> · <a href="http://leafletjs.com" title="A JS library for interactive maps">Leaflet</a>');
        var bounds_group = new L.featureGroup([]);
        function setBounds() {
        }
        function pop_Coastline_0(feature, layer) {
            var popupContent = '<table>\
                    <tr>\
                        <td colspan="2">' + (feature.properties['featurecla'] !== null ? Autolinker.link(String(feature.properties['featurecla'])) : '') + '</td>\
                    </tr>\
                </table>';
            layer.bindPopup(popupContent, {maxHeight: 400});
        }

        function style_Coastline_0_0() {
            return {
                pane: 'pane_Coastline_0',
                opacity: 1,
                color: 'rgba(232,113,141,1.0)',
                dashArray: '',
                lineCap: 'square',
                lineJoin: 'bevel',
                weight: 1.0,
                fillOpacity: 0,
            }
        }
        map.createPane('pane_Coastline_0');
        map.getPane('pane_Coastline_0').style.zIndex = 400;
        map.getPane('pane_Coastline_0').style['mix-blend-mode'] = 'normal';
        var layer_Coastline_0 = new L.geoJson(json_Coastline_0, {
            attribution: '',
            pane: 'pane_Coastline_0',
            onEachFeature: pop_Coastline_0,
            style: style_Coastline_0_0,
        });
        bounds_group.addLayer(layer_Coastline_0);
        map.addLayer(layer_Coastline_0);
        function pop_STPEEZ_1(feature, layer) {
                var popupContent = '<table>\
                        <tr>\
                            <td colspan="2">' + (feature.properties['geoname'] != null ? Autolinker.link(String(feature.properties['geoname'])) : '') + '</td>\
                        </tr>\
                    </table>';
                layer.bindPopup(popupContent, {maxHeight: 400});
            }

            function style_STPEEZ_1_0() {
                return {
                    pane: 'pane_STPEEZ_1',
                    opacity: 1,
                    color: 'rgba(100,152,210,1.0)',
                    dashArray: '',
                    lineCap: 'butt',
                    lineJoin: 'miter',
                    weight: 1.0,
                    fill: true,
                    fillOpacity: 1,
                    fillColor: 'rgba(165,191,221,1.0)',
                }
            }
```

```
122.                map.createPane('pane_STPEEZ_1');
123.                map.getPane('pane_STPEEZ_1').style.zIndex = 401;
124.                map.getPane('pane_STPEEZ_1').style['mix-blend-mode'] = 'normal';
125.                var layer_STPEEZ_1 = new L.geoJson(json_STPEEZ_1, {
126.                    attribution: '',
127.                    pane: 'pane_STPEEZ_1',
128.                    onEachFeature: pop_STPEEZ_1,
129.                    style: style_STPEEZ_1_0,
130.                });
131.                bounds_group.addLayer(layer_STPEEZ_1);
132.                function pop_STPoutline_2(feature, layer) {
133.                    var popupContent = '<table>\
134.                            <tr>\
135.                                <td colspan="2">' + (feature.properties['ID'] !== nul
    l ? Autolinker.link(String(feature.properties['ID'])) : '') + '</td>\
136.                            </tr>\
137.                        </table>';
138.                    layer.bindPopup(popupContent, {maxHeight: 400});
139.                }
140.
141.                function style_STPoutline_2_0() {
142.                    return {
143.                        pane: 'pane_STPoutline_2',
144.                        opacity: 1,
145.                        color: 'rgba(35,35,35,1.0)',
146.                        dashArray: '',
147.                        lineCap: 'square',
148.                        lineJoin: 'bevel',
149.                        weight: 1.0,
150.                        fillOpacity: 0,
151.                    }
152.                }
153.                map.createPane('pane_STPoutline_2');
154.                map.getPane('pane_STPoutline_2').style.zIndex = 402;
155.                map.getPane('pane_STPoutline_2').style['mix-blend-
    mode'] = 'normal';
156.                var layer_STPoutline_2 = new L.geoJson(json_STPoutline_2, {
157.                    attribution: '',
158.                    pane: 'pane_STPoutline_2',
159.                    onEachFeature: pop_STPoutline_2,
160.                    style: style_STPoutline_2_0,
161.                });
162.                bounds_group.addLayer(layer_STPoutline_2);
163.                function pop_STParea_3(feature, layer) {
164.                    var popupContent = '<table>\
165.                            <tr>\
166.                                <td colspan="2">' + (feature.properties['ID'] !== nul
    l ? Autolinker.link(String(feature.properties['ID'])) : '') + '</td>\
167.                            </tr>\
168.                        </table>';
169.                    layer.bindPopup(popupContent, {maxHeight: 400});
170.                }
171.
172.                function style_STParea_3_0() {
173.                    return {
174.                        pane: 'pane_STParea_3',
175.                        stroke: false,
176.                        fill: true,
177.                        fillOpacity: 1,
178.                        fillColor: 'rgba(148,209,128,1.0)',
179.                    }
180.                }
181.                map.createPane('pane_STParea_3');
182.                map.getPane('pane_STParea_3').style.zIndex = 403;
183.                map.getPane('pane_STParea_3').style['mix-blend-mode'] = 'normal';
184.                var layer_STParea_3 = new L.geoJson(json_STParea_3, {
```

```
                    attribution: '',
                    pane: 'pane_STParea_3',
                    onEachFeature: pop_STParea_3,
                    style: style_STParea_3_0,
                });
                bounds_group.addLayer(layer_STParea_3);
                map.addLayer(layer_STParea_3);
                function pop_STPboundaries_4(feature, layer) {
                }

                function style_STPboundaries_4_0() {
                    return {
                        pane: 'pane_STPboundaries_4',
                        opacity: 1,
                        color: 'rgba(0,0,0,1.0)',
                        dashArray: '',
                        lineCap: 'square',
                        lineJoin: 'bevel',
                        weight: 1.0,
                        fillOpacity: 0,
                    }
                }
                map.createPane('pane_STPboundaries_4');
                map.getPane('pane_STPboundaries_4').style.zIndex = 404;
                map.getPane('pane_STPboundaries_4').style['mix-blend-
    mode'] = 'normal';
                var layer_STPboundaries_4 = new L.geoJson(json_STPboundaries_4, {
                    attribution: '',
                    pane: 'pane_STPboundaries_4',
                    onEachFeature: pop_STPboundaries_4,
                    style: style_STPboundaries_4_0,
                });
                bounds_group.addLayer(layer_STPboundaries_4);
                function pop_MOZEEZ_5(feature, layer) {
                    var popupContent = '<table>\
                            <tr>\
                                <td colspan="2">' + (feature.properties['geoname'] !=
    = null ? Autolinker.link(String(feature.properties['geoname'])) : '') + '</td>\
                            </tr>\
                        </table>';
                    layer.bindPopup(popupContent, {maxHeight: 400});
                }

                function style_MOZEEZ_5_0() {
                    return {
                        pane: 'pane_MOZEEZ_5',
                        opacity: 1,
                        color: 'rgba(100,152,210,1.0)',
                        dashArray: '',
                        lineCap: 'butt',
                        lineJoin: 'miter',
                        weight: 1.0,
                        fill: true,
                        fillOpacity: 1,
                        fillColor: 'rgba(165,191,221,1.0)',
                    }
                }
                map.createPane('pane_MOZEEZ_5');
                map.getPane('pane_MOZEEZ_5').style.zIndex = 405;
                map.getPane('pane_MOZEEZ_5').style['mix-blend-mode'] = 'normal';
                var layer_MOZEEZ_5 = new L.geoJson(json_MOZEEZ_5, {
                    attribution: '',
                    pane: 'pane_MOZEEZ_5',
                    onEachFeature: pop_MOZEEZ_5,
                    style: style_MOZEEZ_5_0,
                });
```

```
249.                bounds_group.addLayer(layer_MOZEEZ_5);
250.                function pop_MOZoutline_6(feature, layer) {
251.                    var popupContent = '<table>\
252.                            <tr>\
253.                                <td colspan="2">' + (feature.properties['ID'] !== nul
    l ? Autolinker.link(String(feature.properties['ID'])) : '') + '</td>\
254.                            </tr>\
255.                        </table>';
256.                    layer.bindPopup(popupContent, {maxHeight: 400});
257.                }
258.
259.                function style_MOZoutline_6_0() {
260.                    return {
261.                        pane: 'pane_MOZoutline_6',
262.                        opacity: 1,
263.                        color: 'rgba(0,0,0,1.0)',
264.                        dashArray: '',
265.                        lineCap: 'square',
266.                        lineJoin: 'bevel',
267.                        weight: 1.0,
268.                        fillOpacity: 0,
269.                    }
270.                }
271.                map.createPane('pane_MOZoutline_6');
272.                map.getPane('pane_MOZoutline_6').style.zIndex = 406;
273.                map.getPane('pane_MOZoutline_6').style['mix-blend-
    mode'] = 'normal';
274.                var layer_MOZoutline_6 = new L.geoJson(json_MOZoutline_6, {
275.                    attribution: '',
276.                    pane: 'pane_MOZoutline_6',
277.                    onEachFeature: pop_MOZoutline_6,
278.                    style: style_MOZoutline_6_0,
279.                });
280.                bounds_group.addLayer(layer_MOZoutline_6);
281.                function pop_MOZarea_7(feature, layer) {
282.                    var popupContent = '<table>\
283.                            <tr>\
284.                                <td colspan="2">' + (feature.properties['ID'] !== nul
    l ? Autolinker.link(String(feature.properties['ID'])) : '') + '</td>\
285.                            </tr>\
286.                        </table>';
287.                    layer.bindPopup(popupContent, {maxHeight: 400});
288.                }
289.
290.                function style_MOZarea_7_0() {
291.                    return {
292.                        pane: 'pane_MOZarea_7',
293.                        stroke: false,
294.                        fill: true,
295.                        fillOpacity: 1,
296.                        fillColor: 'rgba(148,209,128,1.0)',
297.                    }
298.                }
299.                map.createPane('pane_MOZarea_7');
300.                map.getPane('pane_MOZarea_7').style.zIndex = 407;
301.                map.getPane('pane_MOZarea_7').style['mix-blend-mode'] = 'normal';
302.                var layer_MOZarea_7 = new L.geoJson(json_MOZarea_7, {
303.                    attribution: '',
304.                    pane: 'pane_MOZarea_7',
305.                    onEachFeature: pop_MOZarea_7,
306.                    style: style_MOZarea_7_0,
307.                });
308.                bounds_group.addLayer(layer_MOZarea_7);
309.                map.addLayer(layer_MOZarea_7);
310.                function pop_MOZboundaries_8(feature, layer) {
311.                }
```

```
312.
313.                function style_MOZboundaries_8_0() {
314.                    return {
315.                        pane: 'pane_MOZboundaries_8',
316.                        opacity: 1,
317.                        color: 'rgba(0,0,0,1.0)',
318.                        dashArray: '',
319.                        lineCap: 'square',
320.                        lineJoin: 'bevel',
321.                        weight: 1.0,
322.                        fillOpacity: 0,
323.                    }
324.                }
325.                map.createPane('pane_MOZboundaries_8');
326.                map.getPane('pane_MOZboundaries_8').style.zIndex = 408;
327.                map.getPane('pane_MOZboundaries_8').style['mix-blend-
    mode'] = 'normal';
328.                var layer_MOZboundaries_8 = new L.geoJson(json_MOZboundaries_8, {
329.                    attribution: '',
330.                    pane: 'pane_MOZboundaries_8',
331.                    onEachFeature: pop_MOZboundaries_8,
332.                    style: style_MOZboundaries_8_0,
333.                });
334.                bounds_group.addLayer(layer_MOZboundaries_8);
335.                function pop_Seagrasspointdata_9(feature, layer) {
336.                    var popupContent = '<table>\
337.                            <tr>\
338.                                <th scope="row">Family</th>\
339.                                <td>' + (feature.properties['FAMILY'] !== null ? Auto
    linker.link(String(feature.properties['FAMILY'])) : '') + '</td>\
340.                            </tr>\
341.                            <tr>\
342.                                <th scope="row">Genus</th>\
343.                                <td>' + (feature.properties['GENUS'] !== null ? Autol
    inker.link(String(feature.properties['GENUS'])) : '') + '</td>\
344.                            </tr>\
345.                            <tr>\
346.                                <th scope="row">Species</th>\
347.                                <td>' + (feature.properties['SPECIES'] !== null ? Aut
    olinker.link(String(feature.properties['SPECIES'])) : '') + '</td>\
348.                            </tr>\
349.                            <tr>\
350.                                <th scope="row">Latitude</th>\
351.                                <td>' + (feature.properties['Latitude'] !== null ? Au
    tolinker.link(String(feature.properties['Latitude'])) : '') + '</td>\
352.                            </tr>\
353.                            <tr>\
354.                                <th scope="row">Longitude</th>\
355.                                <td>' + (feature.properties['Longitude'] !== null ? A
    utolinker.link(String(feature.properties['Longitude'])) : '') + '</td>\
356.                            </tr>\
357.                            <tr>\
358.                                <th scope="row">Area (km²)</th>\
359.                                <td>' + (feature.properties['Area'] !== null ? Autoli
    nker.link(String(feature.properties['Area'])) : '') + '</td>\
360.                            </tr>\
361.                            <tr>\
362.                                <th scope="row">References</th>\
363.                                <td>' + (feature.properties['References'] !== null ?
    Autolinker.link(String(feature.properties['References'])) : '') + '</td>\
364.                            </tr>\
365.                        </table>';
366.                    layer.bindPopup(popupContent, {maxHeight: 400});
367.                }
368.
369.                function style_Seagrasspointdata_9_0() {
```

```
370.                     return {
371.                         pane: 'pane_Seagrasspointdata_9',
372.                         shape: 'diamond',
373.                         radius: 8.8,
374.                         opacity: 1,
375.                         color: 'rgba(50,87,128,1.0)',
376.                         dashArray: '',
377.                         lineCap: 'butt',
378.                         lineJoin: 'miter',
379.                         weight: 2.0,
380.                         fill: true,
381.                         fillOpacity: 1,
382.                         fillColor: 'rgba(72,123,182,1.0)',
383.                     }
384.                 }
385.                 map.createPane('pane_Seagrasspointdata_9');
386.                 map.getPane('pane_Seagrasspointdata_9').style.zIndex = 409;
387.                 map.getPane('pane_Seagrasspointdata_9').style['mix-blend-
    mode'] = 'normal';
388.                 var layer_Seagrasspointdata_9 = new L.geoJson(json_Seagrasspointdata_
    9, {
389.                     attribution: '',
390.                     pane: 'pane_Seagrasspointdata_9',
391.                     onEachFeature: pop_Seagrasspointdata_9,
392.                     pointToLayer: function (feature, latlng) {
393.                         var context = {
394.                             feature: feature,
395.                             variables: {}
396.                         };
397.                         return L.shapeMarker(latlng, style_Seagrasspointdata_9_0(feat
    ure));
398.                     },
399.                 });
400.                 var cluster_Seagrasspointdata_9 = new L.MarkerClusterGroup({showCover
    ageOnHover: false,
401.                     spiderfyDistanceMultiplier: 2});
402.                 cluster_Seagrasspointdata_9.addLayer(layer_Seagrasspointdata_9);
403.
404.                 bounds_group.addLayer(layer_Seagrasspointdata_9);
405.                 function pop_Seagrasspolygondata_10(feature, layer) {
406.                 }
407.
408.                 function style_Seagrasspolygondata_10_0() {
409.                     return {
410.                         pane: 'pane_Seagrasspolygondata_10',
411.                         stroke: false,
412.                         fill: true,
413.                         fillOpacity: 1,
414.                         fillColor: 'rgba(163,205,185,1.0)',
415.                     }
416.                 }
417.                 function style_Seagrasspolygondata_10_1() {
418.                     return {
419.                         pane: 'pane_Seagrasspolygondata_10',
420.                     }
421.                 }
422.                 map.createPane('pane_Seagrasspolygondata_10');
423.                 map.getPane('pane_Seagrasspolygondata_10').style.zIndex = 410;
424.                 map.getPane('pane_Seagrasspolygondata_10').style['mix-blend-
    mode'] = 'normal';
425.                 var layer_Seagrasspolygondata_10 = new L.geoJson.multiStyle(json_Seag
    rasspolygondata_10, {
426.                     attribution: '',
427.                     pane: 'pane_Seagrasspolygondata_10',
428.                     onEachFeature: pop_Seagrasspolygondata_10,
```

```
429.                     styles: [style_Seagrasspolygondata_10_0,style_Seagrasspolygondata
    _10_1,]
430.                 });
431.                 bounds_group.addLayer(layer_Seagrasspolygondata_10);
432.                 function pop_Mangrovepointdata_11(feature, layer) {
433.                     var popupContent = '<table>\
434.                             <tr>\
435.                                 <th scope="row">Location</th>\
436.                                 <td>' + (feature.properties['Name'] !== null ? Autoli
    nker.link(String(feature.properties['Name'])) : '') + '</td>\
437.                             </tr>\
438.                             <tr>\
439.                                 <th scope="row">Latitude</th>\
440.                                 <td>' + (feature.properties['Latitude'] !== null ? Au
    tolinker.link(String(feature.properties['Latitude'])) : '') + '</td>\
441.                             </tr>\
442.                             <tr>\
443.                                 <th scope="row">Longitude</th>\
444.                                 <td>' + (feature.properties['Longitude'] !== null ? A
    utolinker.link(String(feature.properties['Longitude'])) : '') + '</td>\
445.                             </tr>\
446.                             <tr>\
447.                                 <th scope="row">Area (km²)</th>\
448.                                 <td>' + (feature.properties['Area'] !== null ? Autoli
    nker.link(String(feature.properties['Area'])) : '') + '</td>\
449.                             </tr>\
450.                             <tr>\
451.                                 <th scope="row">References</th>\
452.                                 <td>' + (feature.properties['References'] !== null ?
    Autolinker.link(String(feature.properties['References'])) : '') + '</td>\
453.                             </tr>\
454.                         </table>';
455.                     layer.bindPopup(popupContent, {maxHeight: 400});
456.                 }
457.
458.                 function style_Mangrovepointdata_11_0() {
459.                     return {
460.                         pane: 'pane_Mangrovepointdata_11',
461.                         shape: 'diamond',
462.                         radius: 8.8,
463.                         opacity: 1,
464.                         color: 'rgba(61,128,53,1.0)',
465.                         dashArray: '',
466.                         lineCap: 'butt',
467.                         lineJoin: 'miter',
468.                         weight: 2.0,
469.                         fill: true,
470.                         fillOpacity: 1,
471.                         fillColor: 'rgba(84,176,74,1.0)',
472.                     }
473.                 }
474.                 map.createPane('pane_Mangrovepointdata_11');
475.                 map.getPane('pane_Mangrovepointdata_11').style.zIndex = 411;
476.                 map.getPane('pane_Mangrovepointdata_11').style['mix-blend-
    mode'] = 'normal';
477.                 var layer_Mangrovepointdata_11 = new L.geoJson(json_Mangrovepointdata
    _11, {
478.                     attribution: '',
479.                     pane: 'pane_Mangrovepointdata_11',
480.                     onEachFeature: pop_Mangrovepointdata_11,
481.                     pointToLayer: function (feature, latlng) {
482.                         var context = {
483.                             feature: feature,
484.                             variables: {}
485.                         };
```

```
                        return L.shapeMarker(latlng, style_Mangrovepointdata_11_0(feature));
                    },
                });
                var cluster_Mangrovepointdata_11 = new L.MarkerClusterGroup({showCoverageOnHover: false,
                    spiderfyDistanceMultiplier: 2});
                cluster_Mangrovepointdata_11.addLayer(layer_Mangrovepointdata_11);

                bounds_group.addLayer(layer_Mangrovepointdata_11);
                function pop_Mangrovepolygondata_12(feature, layer) {
                }

                var pattern_Mangrovepolygondata_12_0 = new L.StripePattern({
                    weight: 0.3,
                    spaceWeight: 2.0,
                    color: '#4daf4a',
                    opacity: 1.0,
                    spaceOpacity: 0,
                    angle: 225
                });
                pattern_Mangrovepolygondata_12_0.addTo(map);
                function style_Mangrovepolygondata_12_0() {
                    return {
                        pane: 'pane_Mangrovepolygondata_12',
                        stroke: false,
                        fillOpacity: 1,
                        fillPattern: pattern_Mangrovepolygondata_12_0
                    }
                }
                function style_Mangrovepolygondata_12_1() {
                    return {
                        pane: 'pane_Mangrovepolygondata_12',
                        opacity: 1,
                        color: 'rgba(77,175,74,1.0)',
                        dashArray: '',
                        lineCap: 'butt',
                        lineJoin: 'miter',
                        weight: 2.0,
                        fillOpacity: 0,
                    }
                }
                map.createPane('pane_Mangrovepolygondata_12');
                map.getPane('pane_Mangrovepolygondata_12').style.zIndex = 412;
                map.getPane('pane_Mangrovepolygondata_12').style['mix-blend-mode'] = 'normal';
                var layer_Mangrovepolygondata_12 = new L.geoJson.multiStyle(json_Mangrovepolygondata_12, {
                    attribution: '',
                    pane: 'pane_Mangrovepolygondata_12',
                    onEachFeature: pop_Mangrovepolygondata_12,
                    styles: [style_Mangrovepolygondata_12_0,style_Mangrovepolygondata_12_1,]
                });
                bounds_group.addLayer(layer_Mangrovepolygondata_12);
                function pop_Coralpointdata_13(feature, layer) {
                    var popupContent = '<table>\
                            <tr>\
                                <th scope="row">Locality</th>\
                                <td>' + (feature.properties['SITE_NAME'] !== null ? Autolinker.link(String(feature.properties['SITE_NAME'])) : '') + '</td>\
                            </tr>\
                            <tr>\
                                <th scope="row">Latitude</th>\
                                <td>' + (feature.properties['LAT'] !== null ? Autolinker.link(String(feature.properties['LAT'])) : '') + '</td>\
```

```
                            </tr>\
                            <tr>\
                                <th scope="row">Longitude</th>\
                                <td>' + (feature.properties['LON'] !== null ? Autolin
    ker.link(String(feature.properties['LON'])) : '') + '</td>\
                            </tr>\
                            <tr>\
                                <th scope="row">References</th>\
                                <td>' + (feature.properties['SOURCES'] !== null ? Aut
    olinker.link(String(feature.properties['SOURCES'])) : '') + '</td>\
                            </tr>\
                        </table>';
                    layer.bindPopup(popupContent, {maxHeight: 400});
                }

                function style_Coralpointdata_13_0() {
                    return {
                        pane: 'pane_Coralpointdata_13',
                        shape: 'diamond',
                        radius: 8.8,
                        opacity: 1,
                        color: 'rgba(128,17,25,1.0)',
                        dashArray: '',
                        lineCap: 'butt',
                        lineJoin: 'miter',
                        weight: 2.0,
                        fill: true,
                        fillOpacity: 1,
                        fillColor: 'rgba(219,30,42,1.0)',
                    }
                }
                map.createPane('pane_Coralpointdata_13');
                map.getPane('pane_Coralpointdata_13').style.zIndex = 413;
                map.getPane('pane_Coralpointdata_13').style['mix-blend-
    mode'] = 'normal';
                var layer_Coralpointdata_13 = new L.geoJson(json_Coralpointdata_13, {

                    attribution: '',
                    pane: 'pane_Coralpointdata_13',
                    onEachFeature: pop_Coralpointdata_13,
                    pointToLayer: function (feature, latlng) {
                        var context = {
                            feature: feature,
                            variables: {}
                        };
                        return L.shapeMarker(latlng, style_Coralpointdata_13_0(featur
    e));
                    },
                });
                var cluster_Coralpointdata_13 = new L.MarkerClusterGroup({showCoverag
    eOnHover: false,
                    spiderfyDistanceMultiplier: 2});
                cluster_Coralpointdata_13.addLayer(layer_Coralpointdata_13);

                bounds_group.addLayer(layer_Coralpointdata_13);
                function pop_Coralpolygondata_14(feature, layer) {
                }

                function style_Coralpolygondata_14_0() {
                    return {
                        pane: 'pane_Coralpolygondata_14',
                    }
                }
                function style_Coralpolygondata_14_1() {
                    return {
                        pane: 'pane_Coralpolygondata_14',
```

```
                        opacity: 1,
                        color: 'rgba(228,26,28,1.0)',
                        dashArray: '',
                        lineCap: 'butt',
                        lineJoin: 'miter',
                        weight: 1.0,
                        fillOpacity: 0,
                    }
                }
                map.createPane('pane_Coralpolygondata_14');
                map.getPane('pane_Coralpolygondata_14').style.zIndex = 414;
                map.getPane('pane_Coralpolygondata_14').style['mix-blend-
    mode'] = 'normal';
                var layer_Coralpolygondata_14 = new L.geoJson.multiStyle(json_Coralpo
    lygondata_14, {
                    attribution: '',
                    pane: 'pane_Coralpolygondata_14',
                    onEachFeature: pop_Coralpolygondata_14,
                    styles: [style_Coralpolygondata_14_0,style_Coralpolygondata_14_1,
    ]
                });
                bounds_group.addLayer(layer_Coralpolygondata_14);
                function pop_Speciesoccurrences_15(feature, layer) {
                    var popupContent = '<table>\
                            <tr>\
                                <th scope="row">Country</th>\
                                <td>' + (feature.properties['country'] !== null ? Aut
    olinker.link(String(feature.properties['country'])) : '') + '</td>\
                            </tr>\
                            <tr>\
                                <th scope="row">Locality</th>\
                                <td>' + (feature.properties['locality'] !== null ? Au
    tolinker.link(String(feature.properties['locality'])) : '') + '</td>\
                            </tr>\
                            <tr>\
                                <th scope="row">Habitat</th>\
                                <td>' + (feature.properties['habitat'] !== null ? Aut
    olinker.link(String(feature.properties['habitat'])) : '') + '</td>\
                            </tr>\
                            <tr>\
                                <th scope="row">Latitude</th>\
                                <td>' + (feature.properties['decimallatitude'] !== nu
    ll ? Autolinker.link(String(feature.properties['decimallatitude'])) : '') + '</td>\

                            </tr>\
                            <tr>\
                                <th scope="row">Longitude</th>\
                                <td>' + (feature.properties['decimallongitude'] !== n
    ull ? Autolinker.link(String(feature.properties['decimallongitude'])) : '') + '</td>
    \
                            </tr>\
                            <tr>\
                                <th scope="row">Depth (m)</th>\
                                <td>' + (feature.properties['occurrences_depthinmeter
    s'] !== null ? Autolinker.link(String(feature.properties['occurrences_depthinmeters'
    ])) : '') + '</td>\
                            </tr>\
                            <tr>\
                                <th scope="row"> Nr. individuals</th>\
                                <td>' + (feature.properties['occurrences_individualco
    unt'] !== null ? Autolinker.link(String(feature.properties['occurrences_individualco
    unt'])) : '') + '</td>\
                            </tr>\
                            <tr>\
                                <th scope="row">Date</th>\
```

```
656.                                <td>' + (feature.properties['occurrences_eventdate']
    !== null ? Autolinker.link(String(feature.properties['occurrences_eventdate'])) : ''
    ) + '</td>\
657.                            </tr>\
658.                            <tr>\
659.                                <th scope="row">Scientific Name</th>\
660.                                <td>' + (feature.properties['search_scientificname']
    !== null ? Autolinker.link(String(feature.properties['search_scientificname'])) : ''
    ) + '</td>\
661.                            </tr>\
662.                            <tr>\
663.                                <th scope="row">Taxonomy</th>\
664.                                <td>' + (feature.properties['species_taxoninfo'] !==
    null ? Autolinker.link(String(feature.properties['species_taxoninfo'])) : '') + '</t
    d>\
665.                            </tr>\
666.                        </table>';
667.                    layer.bindPopup(popupContent, {maxHeight: 400});
668.                }
669.                function style_Speciesoccurrences_15_0(feature) {
670.                    var context = {
671.                        feature: feature,
672.                        variables: {}
673.                    };
674.                    // Start of if blocks and style check logic
675.                    if (Speciesoccurrences_15rule0_eval_expression(context)) {
676.                      return {
677.                        pane: 'pane_Speciesoccurrences_15',
678.                        opacity: 0.7,
679.                rotationAngle: 0.0,
680.                rotationOrigin: 'center center',
681.                icon: L.icon({
682.                    iconUrl: 'markers/crab-2.svg',
683.                    iconSize: [19.0, 19.0]
684.                }),
685.                    };
686.                        }
687.                        else if (Speciesoccurrences_15rule1_eval_expression(context))
    {
688.                          return {
689.                        pane: 'pane_Speciesoccurrences_15',
690.                        opacity: 0.7,
691.                rotationAngle: 0.0,
692.                rotationOrigin: 'center center',
693.                icon: L.icon({
694.                    iconUrl: 'markers/crab-2.svg',
695.                    iconSize: [21.923074, 21.923074]
696.                }),
697.                    };
698.                        }
699.                        else if (Speciesoccurrences_15rule2_eval_expression(context))
    {
700.                          return {
701.                        pane: 'pane_Speciesoccurrences_15',
702.                        opacity: 0.7,
703.                rotationAngle: 0.0,
704.                rotationOrigin: 'center center',
705.                icon: L.icon({
706.                    iconUrl: 'markers/crab-2.svg',
707.                    iconSize: [24.846148, 24.846148]
708.                }),
709.                    };
710.                        }
711.                        else if (Speciesoccurrences_15rule3_eval_expression(context))
    {
712.                          return {
```

```
713.                         pane: 'pane_Speciesoccurrences_15',
714.                         opacity: 0.7,
715.                 rotationAngle: 0.0,
716.                 rotationOrigin: 'center center',
717.                 icon: L.icon({
718.                     iconUrl: 'markers/crab-2.svg',
719.                     iconSize: [27.769222, 27.769222]
720.                 }),
721.                     };
722.                         }
723.                         else if (Speciesoccurrences_15rule4_eval_expression(context))
    {
724.                           return {
725.                         pane: 'pane_Speciesoccurrences_15',
726.                         opacity: 0.7,
727.                 rotationAngle: 0.0,
728.                 rotationOrigin: 'center center',
729.                 icon: L.icon({
730.                     iconUrl: 'markers/crab-2.svg',
731.                     iconSize: [30.692295999999995, 30.692295999999995]
732.                 }),
733.                     };
734.                         }
735.                         else if (Speciesoccurrences_15rule5_eval_expression(context))
    {
736.                           return {
737.                         pane: 'pane_Speciesoccurrences_15',
738.                         opacity: 0.7,
739.                 rotationAngle: 0.0,
740.                 rotationOrigin: 'center center',
741.                 icon: L.icon({
742.                     iconUrl: 'markers/crab-2.svg',
743.                     iconSize: [33.61537, 33.61537]
744.                 }),
745.                     };
746.                         }
747.                         else if (Speciesoccurrences_15rule6_eval_expression(context))
    {
748.                           return {
749.                         pane: 'pane_Speciesoccurrences_15',
750.                         opacity: 0.7,
751.                 rotationAngle: 0.0,
752.                 rotationOrigin: 'center center',
753.                 icon: L.icon({
754.                     iconUrl: 'markers/crab-2.svg',
755.                     iconSize: [36.538444, 36.538444]
756.                 }),
757.                     };
758.                         }
759.                         else if (Speciesoccurrences_15rule7_eval_expression(context))
    {
760.                           return {
761.                         pane: 'pane_Speciesoccurrences_15',
762.                         opacity: 0.7,
763.                 rotationAngle: 0.0,
764.                 rotationOrigin: 'center center',
765.                 icon: L.icon({
766.                     iconUrl: 'markers/crab-2.svg',
767.                     iconSize: [39.46148, 39.46148]
768.                 }),
769.                     };
770.                         }
771.                         else if (Speciesoccurrences_15rule8_eval_expression(context))
    {
772.                           return {
773.                         pane: 'pane_Speciesoccurrences_15',
```

```
774.                         opacity: 0.7,
775.                 rotationAngle: 0.0,
776.                 rotationOrigin: 'center center',
777.                 icon: L.icon({
778.                     iconUrl: 'markers/crab-2.svg',
779.                     iconSize: [42.38444, 42.38444]
780.                 }),
781.                     };
782.                         }
783.                         else if (Speciesoccurrences_15rule9_eval_expression(context))
    {
784.                           return {
785.                         pane: 'pane_Speciesoccurrences_15',
786.                         opacity: 0.7,
787.                 rotationAngle: 0.0,
788.                 rotationOrigin: 'center center',
789.                 icon: L.icon({
790.                     iconUrl: 'markers/crab-2.svg',
791.                     iconSize: [45.307779999999994, 45.307779999999994]
792.                 }),
793.                     };
794.                         }
795.                         else if (Speciesoccurrences_15rule10_eval_expression(context)
    ) {
796.                           return {
797.                         pane: 'pane_Speciesoccurrences_15',
798.                         opacity: 0.7,
799.                 rotationAngle: 0.0,
800.                 rotationOrigin: 'center center',
801.                 icon: L.icon({
802.                     iconUrl: 'markers/crab-2.svg',
803.                     iconSize: [48.23074, 48.23074]
804.                 }),
805.                     };
806.                         }
807.                         else if (Speciesoccurrences_15rule11_eval_expression(context)
    ) {
808.                           return {
809.                         pane: 'pane_Speciesoccurrences_15',
810.                         opacity: 0.7,
811.                 rotationAngle: 0.0,
812.                 rotationOrigin: 'center center',
813.                 icon: L.icon({
814.                     iconUrl: 'markers/crab-2.svg',
815.                     iconSize: [51.15369999999999, 51.15369999999999]
816.                 }),
817.                     };
818.                         }
819.                         else if (Speciesoccurrences_15rule12_eval_expression(context)
    ) {
820.                           return {
821.                         pane: 'pane_Speciesoccurrences_15',
822.                         opacity: 0.7,
823.                 rotationAngle: 0.0,
824.                 rotationOrigin: 'center center',
825.                 icon: L.icon({
826.                     iconUrl: 'markers/crab-2.svg',
827.                     iconSize: [54.07704, 54.07704]
828.                 }),
829.                     };
830.                         }
831.                         else if (Speciesoccurrences_15rule13_eval_expression(context)
    ) {
832.                           return {
833.                         pane: 'pane_Speciesoccurrences_15',
834.                         opacity: 0.7,
```

```
                rotationAngle: 0.0,
                rotationOrigin: 'center center',
                icon: L.icon({
                    iconUrl: 'markers/crab-2.svg',
                    iconSize: [57.0, 57.0]
                }),
                    };
                        }
                        else if (Speciesoccurrences_15rule14_eval_expression(context)
    ) {
                          return {
                        pane: 'pane_Speciesoccurrences_15',
                        opacity: 0.7,
                rotationAngle: 0.0,
                rotationOrigin: 'center center',
                icon: L.icon({
                    iconUrl: 'markers/d354e30bb05d7f183d0c95ece6ed78cf.svg',
                    iconSize: [19.0, 19.0]
                }),
                    };
                        }
                        else if (Speciesoccurrences_15rule15_eval_expression(context)
    ) {
                          return {
                        pane: 'pane_Speciesoccurrences_15',
                        opacity: 0.7,
                rotationAngle: 0.0,
                rotationOrigin: 'center center',
                icon: L.icon({
                    iconUrl: 'markers/d354e30bb05d7f183d0c95ece6ed78cf.svg',
                    iconSize: [21.923074, 21.923074]
                }),
                    };
                        }
                        else if (Speciesoccurrences_15rule16_eval_expression(context)
    ) {
                          return {
                        pane: 'pane_Speciesoccurrences_15',
                        opacity: 0.7,
                rotationAngle: 0.0,
                rotationOrigin: 'center center',
                icon: L.icon({
                    iconUrl: 'markers/d354e30bb05d7f183d0c95ece6ed78cf.svg',
                    iconSize: [24.846148, 24.846148]
                }),
                    };
                        }
                        else if (Speciesoccurrences_15rule17_eval_expression(context)
    ) {
                          return {
                        pane: 'pane_Speciesoccurrences_15',
                        opacity: 0.7,
                rotationAngle: 0.0,
                rotationOrigin: 'center center',
                icon: L.icon({
                    iconUrl: 'markers/d354e30bb05d7f183d0c95ece6ed78cf.svg',
                    iconSize: [27.769222, 27.769222]
                }),
                    };
                        }
                        else if (Speciesoccurrences_15rule18_eval_expression(context)
    ) {
                          return {
                        pane: 'pane_Speciesoccurrences_15',
                        opacity: 0.7,
                rotationAngle: 0.0,
```

```
896.                 rotationOrigin: 'center center',
897.                 icon: L.icon({
898.                     iconUrl: 'markers/d354e30bb05d7f183d0c95ece6ed78cf.svg',
899.                     iconSize: [30.692295999999995, 30.692295999999995]
900.                 }),
901.                     };
902.                         }
903.                         else if (Speciesoccurrences_15rule19_eval_expression(context)
    ) {
904.                           return {
905.                         pane: 'pane_Speciesoccurrences_15',
906.                         opacity: 0.7,
907.                 rotationAngle: 0.0,
908.                 rotationOrigin: 'center center',
909.                 icon: L.icon({
910.                     iconUrl: 'markers/d354e30bb05d7f183d0c95ece6ed78cf.svg',
911.                     iconSize: [33.61537, 33.61537]
912.                 }),
913.                     };
914.                         }
915.                         else if (Speciesoccurrences_15rule20_eval_expression(context)
    ) {
916.                           return {
917.                         pane: 'pane_Speciesoccurrences_15',
918.                         opacity: 0.7,
919.                 rotationAngle: 0.0,
920.                 rotationOrigin: 'center center',
921.                 icon: L.icon({
922.                     iconUrl: 'markers/d354e30bb05d7f183d0c95ece6ed78cf.svg',
923.                     iconSize: [36.538444, 36.538444]
924.                 }),
925.                     };
926.                         }
927.                         else if (Speciesoccurrences_15rule21_eval_expression(context)
    ) {
928.                           return {
929.                         pane: 'pane_Speciesoccurrences_15',
930.                         opacity: 0.7,
931.                 rotationAngle: 0.0,
932.                 rotationOrigin: 'center center',
933.                 icon: L.icon({
934.                     iconUrl: 'markers/d354e30bb05d7f183d0c95ece6ed78cf.svg',
935.                     iconSize: [39.46148, 39.46148]
936.                 }),
937.                     };
938.                         }
939.                         else if (Speciesoccurrences_15rule22_eval_expression(context)
    ) {
940.                           return {
941.                         pane: 'pane_Speciesoccurrences_15',
942.                         opacity: 0.7,
943.                 rotationAngle: 0.0,
944.                 rotationOrigin: 'center center',
945.                 icon: L.icon({
946.                     iconUrl: 'markers/d354e30bb05d7f183d0c95ece6ed78cf.svg',
947.                     iconSize: [42.38444, 42.38444]
948.                 }),
949.                     };
950.                         }
951.                         else if (Speciesoccurrences_15rule23_eval_expression(context)
    ) {
952.                           return {
953.                         pane: 'pane_Speciesoccurrences_15',
954.                         opacity: 0.7,
955.                 rotationAngle: 0.0,
956.                 rotationOrigin: 'center center',
```

```
957.                icon: L.icon({
958.                    iconUrl: 'markers/d354e30bb05d7f183d0c95ece6ed78cf.svg',
959.                    iconSize: [45.307779999999994, 45.307779999999994]
960.                }),
961.                    };
962.                        }
963.                        else if (Speciesoccurrences_15rule24_eval_expression(context)
    ) {
964.                          return {
965.                        pane: 'pane_Speciesoccurrences_15',
966.                        opacity: 0.7,
967.                rotationAngle: 0.0,
968.                rotationOrigin: 'center center',
969.                icon: L.icon({
970.                    iconUrl: 'markers/d354e30bb05d7f183d0c95ece6ed78cf.svg',
971.                    iconSize: [48.23074, 48.23074]
972.                }),
973.                    };
974.                        }
975.                        else if (Speciesoccurrences_15rule25_eval_expression(context)
    ) {
976.                          return {
977.                        pane: 'pane_Speciesoccurrences_15',
978.                        opacity: 0.7,
979.                rotationAngle: 0.0,
980.                rotationOrigin: 'center center',
981.                icon: L.icon({
982.                    iconUrl: 'markers/d354e30bb05d7f183d0c95ece6ed78cf.svg',
983.                    iconSize: [51.15369999999999, 51.15369999999999]
984.                }),
985.                    };
986.                        }
987.                        else if (Speciesoccurrences_15rule26_eval_expression(context)
    ) {
988.                          return {
989.                        pane: 'pane_Speciesoccurrences_15',
990.                        opacity: 0.7,
991.                rotationAngle: 0.0,
992.                rotationOrigin: 'center center',
993.                icon: L.icon({
994.                    iconUrl: 'markers/d354e30bb05d7f183d0c95ece6ed78cf.svg',
995.                    iconSize: [54.07704, 54.07704]
996.                }),
997.                    };
998.                        }
999.                        else if (Speciesoccurrences_15rule27_eval_expression(context)
    ) {
1000.                         return {
1001.                       pane: 'pane_Speciesoccurrences_15',
1002.                       opacity: 0.7,
1003.               rotationAngle: 0.0,
1004.               rotationOrigin: 'center center',
1005.               icon: L.icon({
1006.                   iconUrl: 'markers/d354e30bb05d7f183d0c95ece6ed78cf.svg',
1007.                   iconSize: [57.0, 57.0]
1008.               }),
1009.                   };
1010.                       }
1011.                       else if (Speciesoccurrences_15rule28_eval_expression(context)
    ) {
1012.                         return {
1013.                       pane: 'pane_Speciesoccurrences_15',
1014.                       opacity: 0.7,
1015.               rotationAngle: 0.0,
1016.               rotationOrigin: 'center center',
1017.               icon: L.icon({
```

```
1018.                 iconUrl: 'markers/crayfish.svg',
1019.                 iconSize: [19.0, 19.0]
1020.             }),
1021.                 };
1022.                     }
1023.                     else if (Speciesoccurrences_15rule29_eval_expression(context)
    ) {
1024.                       return {
1025.                     pane: 'pane_Speciesoccurrences_15',
1026.                     opacity: 0.7,
1027.             rotationAngle: 0.0,
1028.             rotationOrigin: 'center center',
1029.             icon: L.icon({
1030.                 iconUrl: 'markers/crayfish.svg',
1031.                 iconSize: [21.923074, 21.923074]
1032.             }),
1033.                 };
1034.                     }
1035.                     else if (Speciesoccurrences_15rule30_eval_expression(context)
    ) {
1036.                       return {
1037.                     pane: 'pane_Speciesoccurrences_15',
1038.                     opacity: 0.7,
1039.             rotationAngle: 0.0,
1040.             rotationOrigin: 'center center',
1041.             icon: L.icon({
1042.                 iconUrl: 'markers/crayfish.svg',
1043.                 iconSize: [24.846148, 24.846148]
1044.             }),
1045.                 };
1046.                     }
1047.                     else if (Speciesoccurrences_15rule31_eval_expression(context)
    ) {
1048.                       return {
1049.                     pane: 'pane_Speciesoccurrences_15',
1050.                     opacity: 0.7,
1051.             rotationAngle: 0.0,
1052.             rotationOrigin: 'center center',
1053.             icon: L.icon({
1054.                 iconUrl: 'markers/crayfish.svg',
1055.                 iconSize: [27.769222, 27.769222]
1056.             }),
1057.                 };
1058.                     }
1059.                     else if (Speciesoccurrences_15rule32_eval_expression(context)
    ) {
1060.                       return {
1061.                     pane: 'pane_Speciesoccurrences_15',
1062.                     opacity: 0.7,
1063.             rotationAngle: 0.0,
1064.             rotationOrigin: 'center center',
1065.             icon: L.icon({
1066.                 iconUrl: 'markers/crayfish.svg',
1067.                 iconSize: [30.692295999999995, 30.692295999999995]
1068.             }),
1069.                 };
1070.                     }
1071.                     else if (Speciesoccurrences_15rule33_eval_expression(context)
    ) {
1072.                       return {
1073.                     pane: 'pane_Speciesoccurrences_15',
1074.                     opacity: 0.7,
1075.             rotationAngle: 0.0,
1076.             rotationOrigin: 'center center',
1077.             icon: L.icon({
1078.                 iconUrl: 'markers/crayfish.svg',
```

```
1079.                 iconSize: [33.61537, 33.61537]
1080.             }),
1081.                 };
1082.                     }
1083.                     else if (Speciesoccurrences_15rule34_eval_expression(context)
    ) {
1084.                       return {
1085.                     pane: 'pane_Speciesoccurrences_15',
1086.                     opacity: 0.7,
1087.             rotationAngle: 0.0,
1088.             rotationOrigin: 'center center',
1089.             icon: L.icon({
1090.                 iconUrl: 'markers/crayfish.svg',
1091.                 iconSize: [36.538444, 36.538444]
1092.             }),
1093.                 };
1094.                     }
1095.                     else if (Speciesoccurrences_15rule35_eval_expression(context)
    ) {
1096.                       return {
1097.                     pane: 'pane_Speciesoccurrences_15',
1098.                     opacity: 0.7,
1099.             rotationAngle: 0.0,
1100.             rotationOrigin: 'center center',
1101.             icon: L.icon({
1102.                 iconUrl: 'markers/crayfish.svg',
1103.                 iconSize: [39.46148, 39.46148]
1104.             }),
1105.                 };
1106.                     }
1107.                     else if (Speciesoccurrences_15rule36_eval_expression(context)
    ) {
1108.                       return {
1109.                     pane: 'pane_Speciesoccurrences_15',
1110.                     opacity: 0.7,
1111.             rotationAngle: 0.0,
1112.             rotationOrigin: 'center center',
1113.             icon: L.icon({
1114.                 iconUrl: 'markers/crayfish.svg',
1115.                 iconSize: [42.38444, 42.38444]
1116.             }),
1117.                 };
1118.                     }
1119.                     else if (Speciesoccurrences_15rule37_eval_expression(context)
    ) {
1120.                       return {
1121.                     pane: 'pane_Speciesoccurrences_15',
1122.                     opacity: 0.7,
1123.             rotationAngle: 0.0,
1124.             rotationOrigin: 'center center',
1125.             icon: L.icon({
1126.                 iconUrl: 'markers/crayfish.svg',
1127.                 iconSize: [45.307779999999994, 45.307779999999994]
1128.             }),
1129.                 };
1130.                     }
1131.                     else if (Speciesoccurrences_15rule38_eval_expression(context)
    ) {
1132.                       return {
1133.                     pane: 'pane_Speciesoccurrences_15',
1134.                     opacity: 0.7,
1135.             rotationAngle: 0.0,
1136.             rotationOrigin: 'center center',
1137.             icon: L.icon({
1138.                 iconUrl: 'markers/crayfish.svg',
1139.                 iconSize: [48.23074, 48.23074]
```

```
1140.                 }),
1141.                     };
1142.                         }
1143.                         else if (Speciesoccurrences_15rule39_eval_expression(context)
    ) {
1144.                           return {
1145.                         pane: 'pane_Speciesoccurrences_15',
1146.                         opacity: 0.7,
1147.                 rotationAngle: 0.0,
1148.                 rotationOrigin: 'center center',
1149.                 icon: L.icon({
1150.                     iconUrl: 'markers/crayfish.svg',
1151.                     iconSize: [51.15369999999999, 51.15369999999999]
1152.                 }),
1153.                     };
1154.                         }
1155.                         else if (Speciesoccurrences_15rule40_eval_expression(context)
    ) {
1156.                           return {
1157.                         pane: 'pane_Speciesoccurrences_15',
1158.                         opacity: 0.7,
1159.                 rotationAngle: 0.0,
1160.                 rotationOrigin: 'center center',
1161.                 icon: L.icon({
1162.                     iconUrl: 'markers/crayfish.svg',
1163.                     iconSize: [54.07704, 54.07704]
1164.                 }),
1165.                     };
1166.                         }
1167.                         else if (Speciesoccurrences_15rule41_eval_expression(context)
    ) {
1168.                           return {
1169.                         pane: 'pane_Speciesoccurrences_15',
1170.                         opacity: 0.7,
1171.                 rotationAngle: 0.0,
1172.                 rotationOrigin: 'center center',
1173.                 icon: L.icon({
1174.                     iconUrl: 'markers/crayfish.svg',
1175.                     iconSize: [57.0, 57.0]
1176.                 }),
1177.                     };
1178.                         }
1179.                     else {
1180.                         return {fill: false, stroke: false};
1181.                     }
1182.                 }
1183.                 map.createPane('pane_Speciesoccurrences_15');
1184.                 map.getPane('pane_Speciesoccurrences_15').style.zIndex = 415;
1185.                 map.getPane('pane_Speciesoccurrences_15').style['mix-blend-
    mode'] = 'normal';
1186.                 var layer_Speciesoccurrences_15 = new L.geoJson(json_Speciesoccurrenc
    es_15, {
1187.                     attribution: '',
1188.                     pane: 'pane_Speciesoccurrences_15',
1189.                     onEachFeature: pop_Speciesoccurrences_15,
1190.                     pointToLayer: function (feature, latlng) {
1191.                         var context = {
1192.                             feature: feature,
1193.                             variables: {}
1194.                         };
1195.                         return L.marker(latlng, style_Speciesoccurrences_15_0(feature
    ));
1196.                     },
1197.                 });
1198.                 var cluster_Speciesoccurrences_15 = new L.MarkerClusterGroup({showCov
    erageOnHover: false,
```

```
1199.                 spiderfyDistanceMultiplier: 2});
1200.             cluster_Speciesoccurrences_15.addLayer(layer_Speciesoccurrences_15);

1201.
1202.             bounds_group.addLayer(layer_Speciesoccurrences_15);
1203.             cluster_Speciesoccurrences_15.addTo(map);
1204.             var osmGeocoder = new L.Control.Geocoder({
1205.                 collapsed: true,
1206.                 position: 'topleft',
1207.                 text: 'Search',
1208.                 title: 'Testing'
1209.             }).addTo(map);
1210.             document.getElementsByClassName('leaflet-control-geocoder-icon')[0]
1211.             .className += ' fa fa-search';
1212.             document.getElementsByClassName('leaflet-control-geocoder-icon')[0]
1213.             .title += 'Search for a place';
1214.             var baseMaps = {};
1215.             L.control.layers(baseMaps,{'Species occurrences<br/><table><tr><td st
    yle="text-
    align: center;"><img src="legend/Speciesoccurrences_15_0.png";"><img src="legend/Spe
    ciesoccurrences_15_28.png";"><img src="legend/Speciesoccurrences_15_14.png" /></td><
    td></td></tr><tr><td style="text-
    align: center;"></td><td></td></tr></table>': cluster_Speciesoccurrences_15,'<img sr
    c="legend/Coralpolygondata_14.png" /> Coral polygon-
    data': layer_Coralpolygondata_14,'<img src="legend/Coralpointdata_13.png" /> Coral p
    oint-
    data': cluster_Coralpointdata_13,'<img src="legend/Mangrovepolygondata_12.png" /> Ma
    ngrove polygon-
    data': layer_Mangrovepolygondata_12,'<img src="legend/Mangrovepointdata_11.png" /> M
    angrove point-
    data': cluster_Mangrovepointdata_11,'<img src="legend/Seagrasspolygondata_10.png" />
     Seagrass polygon-
    data': layer_Seagrasspolygondata_10,'<img src="legend/Seagrasspointdata_9.png" /> Se
    agrass point-
    data': cluster_Seagrasspointdata_9,'<img src="legend/MOZboundaries_8.png" /> MOZ bou
    ndaries': layer_MOZboundaries_8,'<img src="legend/MOZarea_7.png" /> MOZ area': layer
    _MOZarea_7,'<img src="legend/MOZoutline_6.png" /> MOZ outline': layer_MOZoutline_6,'
    <img src="legend/MOZEEZ_5.png" /> MOZ EEZ': layer_MOZEEZ_5,'<img src="legend/STPboun
    daries_4.png" /> STP boundaries': layer_STPboundaries_4,'<img src="legend/STParea_3.
    png" /> STP area': layer_STParea_3,'<img src="legend/STPoutline_2.png" /> STP outlin
    e': layer_STPoutline_2,'<img src="legend/STPEEZ_1.png" /> STP EEZ': layer_STPEEZ_1,'
    <img src="legend/Coastline_0.png" /> Coastline': layer_Coastline_0,}).addTo(map);
1216.             setBounds();
1217.             map.addControl(new L.Control.Search({
1218.                 layer: cluster_Speciesoccurrences_15,
1219.                 initial: false,
1220.                 hideMarkerOnCollapse: true,
1221.                 propertyName: 'search_scientificname'}));
1222.             document.getElementsByClassName('search-button')[0].className +=
1223.              ' fa fa-binoculars';
1224.             </script>
1225.         </body>
1226.     </html>
```